User manual

# UM EN ILC 1XX

Installing and operating the ILC 130 ETH, ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, and ILC 190 ETH 2TX Inline controllers

PHOENIX CONTACT
INSPIRING INNOVATIONS

# User manual

# Installing and operating the ILC 130 ETH, ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, and ILC 190 ETH 2TX Inline controllers

2012-01-26

Designation: UM EN ILC 1XX

Revision: 06

Order No.: —

This user manual is valid for:

| Designation | Revision (HW) or later | Version (FW) or later | Order No. |
|---|---|---|---|
| ILC 130 ETH | 01 | 3.01 | 2988803 |
| ILC 150 ETH | 00 | 2.00 | 2985330 |
| ILC 155 ETH | 01 | 2.04 | 2988188 |
| ILC 170 ETH 2TX | 01 | 3.00 | 2916532 |
| ILC 190 ETH 2TX | 00 | 3.70 | 2700527 |

# Please observe the following notes

### User group of this manual

The use of products described in this manual is oriented exclusively to qualified electricians or persons instructed by them, who are familiar with applicable standards and other regulations regarding electrical engineering and, in particular, the relevant safety concepts.

### Explanation of symbols used and signal words

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety measures that follow this symbol to avoid possible injury or death.

There are three different categories of personal injury that are indicated with a signal word.

**DANGER** This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

**WARNING** This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

**CAUTION** This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

This symbol together with the signal word **NOTE** and the accompanying text alert the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

This symbol and the accompanying text provide the reader with additional information or refer to detailed sources of information.

### How to contact us

**Internet**

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:
www.phoenixcontact.com

Make sure you always use the latest documentation.
It can be downloaded at:
www.phoenixcontact.net/catalog

**Subsidiaries**

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.
Subsidiary contact information is available at www.phoenixcontact.com.

**Published by**

PHOENIX CONTACT GmbH & Co. KG
Flachsmarktstraße 8
32825 Blomberg
GERMANY

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:
tecdoc@phoenixcontact.com

### General terms and conditions of use for technical documentation

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

This manual, including all illustrations contained herein, is copyright protected. Any changes to the contents or the publication of extracts of this document is prohibited.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.

# Table of contents


1 Introduction ..... 1-1

1.1 Purpose of this manual ..... 1-1

1.2 Hardware and software requirements ..... 1-1

2 Description of the Inline controller ..... 2-1

2.1 General description of the Inline controller ..... 2-1

2.2 Possible fields of application of the Inline controller ..... 2-3

2.2.1 ILC 130 ETH ..... 2-3

2.2.2 ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, and ILC 190 ETH 2TX ..... 2-3

2.2.3 ILC 170 ETH 2TX and ILC 190 ETH 2TX as a PROFINET IO device in a PROFINET network ..... 2-5

2.2.4 Applicative system redundancy with ILC 170 ETH 2TX or ILC 190 ETH 2TX ..... 2-6

2.3 Notes on using the Inline controller (ILC 150 ETH/ILC 155 ETH) in potentially explosive areas ..... 2-7

2.4 Unpacking the Inline controller ..... 2-8

2.5 Connection and operating elements ..... 2-9

2.6 Diagnostic and status indicators ..... 2-11

2.7 Mode selector switch ..... 2-14

2.8 Reset button (concealed) ..... 2-15

2.9 Parameterization memory ..... 2-15

2.10 Internal basic circuit diagram ..... 2-17

2.11 Mounting and removing the Inline controller ..... 2-18

2.12 Communication paths ..... 2-21

2.12.1 Ethernet ..... 2-22

2.12.2 Serial PRG interface (mini-DIN female connector) ..... 2-23

2.12.3 Serial PRG interface - function blocks ..... 2-24

2.13 INTERBUS ..... 2-25

2.13.1 Local bus ..... 2-25

2.13.2 Remote bus ..... 2-25

2.14 Power supply ..... 2-26

2.14.1 Sizing of the power supply ..... 2-26

2.14.2 Power supply connection ..... 2-27

2.14.3 24 V segment supply/24 V main supply ..... 2-29

2.14.4 24 V segment supply ..... 2-29

2.14.5 24 V main voltage ..... 2-29

2.14.6 24 V ILC supply ..... 2-29

2.14.7 Jumpers ..... 2-29

2.15 Digital inputs and outputs ..... 2-30

3 The Inline controller under PC Worx/PC Worx Express ..... 3-1

3.1 Software version ..... 3-1

3.2 Assigning the IP address for the controller/ BootP server ..... 3-2

3.3 ILC 170 ETH 2TX (ILC 190 ETH 2TX) as a PROFINET IO device ..... 3-5

3.4 Setting the realtime clock under PC Worx Express ..... 3-11

3.5 Download changes ..... 3-11

3.6 Parameterization memory and Internet Explorer ..... 3-12

3.6.1 Internet Explorer FTP function ..... 3-13

3.6.2 Activating/deactivating the FTP server ..... 3-13

3.6.3 Activating/deactivating the HTTP server ..... 3-14

3.7 Activating/deactivating the journaling function ..... 3-15

3.8 Function blocks for handling files in the parameterization memory ..... 3-16

3.9 Function blocks for Ethernet communication ..... 3-17

3.10 Function blocks for PCP communication ..... 3-18

3.11 Alignment ..... 3-19

4 System variables and status information ..... 4-1

4.1 General information ..... 4-1

4.2 Status register for local digital inputs and outputs ..... 4-1

4.3 Diagnostic status register ..... 4-2

4.4 Diagnostic parameter register ..... 4-3

4.5 PROFINET register (ILC 170 ETH 2TX and ILC 190 ETH 2TX) ..... 4-4

4.6 IEC -61131 runtime system ..... 4-5

4.7 Control processor ..... 4-6

4.8 Battery, realtime clock ..... 4-7

4.9 Power supplies ..... 4-7

4.10 Mode selector switch ..... 4-7

4.11 System time ..... 4-7

5 Technical data and ordering data ..... 5-1

5.1 Technical data ..... 5-1

5.2 Ordering data ..... 5-8

5.2.1 Modules ..... 5-8

5.2.2 Accessories ..... 5-8

5.2.3 Software ..... 5-8

5.2.4 Documentation ..... 5-8

A Appendix: Service and maintenance ..... A-1

A 1 Error causes and remedies ..... A-1

A 2 Updating the Inline controller firmware ..... A-1

A 3 Connecting unshielded cables ..... A-2

B Index ..... B-1

# 1 Introduction

## 1.1 Purpose of this manual

This manual helps you to start up and operate the following Inline controllers:

– ILC 130 ETH,
– ILC 150 ETH,
– ILC 155 ETH,
– ILC 170 ETH 2TX or
– ILC 190 ETH 2TX.

## 1.2 Hardware and software requirements

| HW/SW | Description | | | | |
|---|---|---|---|---|---|
| Inline controller | **ILC 130 ETH** | **ILC 150 ETH** | **ILC 155 ETH** | **ILC 170 ETH 2TX** | **ILC 190 ETH 2TX** |
| Parameterization memory, plug-in | No | No | No | For the ordering data, see Section “Accessories” on page 5-8 | For the ordering data, see Section “Accessories” on page 5-8 |
| Ethernet cable | Ethernet cable for connecting the Inline controller to a PC | | | | |
| Connecting cable | Connecting cable for connecting the Inline controller to a PC (V.24 (RS-232) cable, optional) | | | | |
| Automation software versions (Service Pack = SP) | | | | | |
| PC Worx | ≥ 5.20 SP3 **** | ≥ 5.10 SP 1 * | ≥ 5.20 ** | ≥ 5.20 SP 2 *** | ≥ 6.10 SP 1 ***** |
| PC Worx Express | ≥ 5.20 SP 3 **** | ≥ 5.20 ** | ≥ 5.20 ** | ≥ 5.20 SP 2 *** | ≥ 6.10 SP 1 *** |

* Part of the AUTOMATIONWORX Software Suite 2007 1.30 Service Pack 1

** Part of the AUTOMATIONWORX Software Suite 2008 1.40

*** Part of the AUTOMATIONWORX Software Suite 2008 1.40 Service Pack 2

**** Part of the AUTOMATIONWORX Software Suite 2008 1.40 Service Pack 3

***** Part of the AUTOMATIONWORX Software Suite 2007 1.60 Service Pack 1

For the ordering data for hardware, software, and additional documentation, please refer to Section “Technical data and ordering data” on page 5-1.

**PROFINET IO device functions**

**ILC 170 ETH 2TX and ILC 190 ETH 2TX**

Please observe that the PROFINET IO device functions of these controllers are only available with the following software versions or later:

| | |
|---|---|
| ILC 170 ETH 2TX | PC Worx Version 6.00 Service Pack 2 or later (part of the AUTOMATIONWORX Software Suite 2009 1.50 Service Pack 2) |
| ILC 190 ETH 2TX | PC Worx Version 6.10 Service Pack 1 or later (part of the AUTOMATIONWORX Software Suite 2010 1.60 Service Pack 1) |

The PC Worx Express software does not support this function.

# 2 Description of the Inline controller

## 2.1 General description of the Inline controller

The Inline controller is a compact controller with integrated Ethernet and INTERBUS connections.

The ILC 130 ETH, ILC 150 ETH, ILC 155 ETH,ILC 170 ETH 2TX, and ILC 190 ETH 2TX Inline controllers have the same appearance and numerous identical functions.

The main difference lies in the varying memory sizes, which are available to the user.

The ILC 150 ETH and ILC 155 ETH Inline controllers are approved for use in zone 2 potentially explosive areas. Observe the notes in Section 2.3 on page 2-7.

The ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers also have a second Ethernet interface and a slot for an SD memory card (parameterization memory).

In the following, the term Inline controller is used in general. Differences between the various controllers are particularly mentioned where necessary.

For additional information about the different memory sizes, please refer to Section 5.1, “Technical data” from page 5-1 onwards.

**IEC 61131 control performance**

The Inline controller is continuously configured and programmed in accordance with IEC 61131 using the automation software PC Worx. PC Worx can be operated via the network (Ethernet). The powerful processor can be programmed in all five IEC 61131 programming languages and ensures quick control task processing.

**Integrated Ethernet connection**

The integrated Ethernet connection (using twisted pair cable) provides for the Ethernet connectivity. Throughout the Ethernet network, the Inline controller can be accessed via Ethernet and TCP/IP or UDP/IP. A standardized Ethernet interface is available for each of the ILC 130 ETH, ILC 150 ETH, and ILC 155 ETH Inline controllers. The ILC 170 ETH 2TX and ILC 190 ETH 2TX controllers have two fully implemented Ethernet connections, which are switched inside the device.

Integrated communication functions enable direct and effective data exchange via Ethernet. The Ethernet network provides universal options for communicating with the Inline controller. Using the send and receive communication blocks, information, e.g.,necessary coupling variables, can be exchanged between Inline controllers via Ethernet. This enables distributed, modular automation solutions to be configured.

The existing IEC 61131-5 blocks have thus been extended to include a transparent TCP/IP mode and a transparent UDP/IP mode.

When using the AX OPC server provided in the AUTOMATIONWORX Software Suite Version 1.30 or later, Inline controller data is available in the Ethernet network in a standardized format and can be used for the different visualization Inline packages.

**PROFINET IO device functions (ILC 170 ETH 2TX and ILC 190 ETH 2TX)**

The PROFINET protocol can additionally be used via the Ethernet interfaces of the ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers. In this case, the Inline controllers can be used as a PROFINET IO device. This function is only available with the following device and software versions or later:

| | |
|---|---|
| ILC 170 ETH 2TX | Hardware "02"/Firmware "3.5x" or later, together with the PC Worx software, Version 6.00 Service Pack 2 or later |
| ILC 190 ETH 2TX | Hardware "00"/Firmware "3.70" or later, together with the PC Worx software, Version 6.10 Service Pack 1 or later |

Please refer to Section “ILC 170 ETH 2TX (ILC 190 ETH 2TX) as a PROFINET IO device” on page 3-5 for information on how to configure the ILC 170 ETH 2TX or ILC 190 ETH 2TX Inline controllers as a PROFINET IO device with the PC Worx software.

**Integrated INTERBUS connection**

An Inline local bus as well as an INTERBUS remote bus can be connected via the INTERBUS connection. In this way you can create a complete INTERBUS system (maximum of 4 remote bus levels) using the Inline controller as the distributed control system.

The I/O level is connected to the Inline controller using INTERBUS.

Please note that the ILC 130 ETH does not support connection of the INTERBUS remote bus.

**V.24 (RS-232) interface**

This interface can be used to **either** assign the IP address of the Inline controller and to access the controller using the Diag+ diagnostic tool **or** to communicate with serial I/O devices via function blocks.

The Inline controller cannot be programmed via the V.24 (RS-232) interface.

**Parameterization memory/SD card (ILC 170 ETH 2TX and ILC 190 ETH 2TX)**

The ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers have a plug-in parameterization memory in the form of an SD card. This memory can be used to save programs and configurations, which belong to your project, e.g., the visualization project.

The parameterization memory is not supplied as standard with the ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers. Use only a parameterization memory provided by Phoenix Contact (for the ordering data, see Section “Accessories” on page 5-8).

**NOTE: Parameterization memory (SD card) – formatting note**

The SD card is preformatted and designed for use with Phoenix Contact devices. Ensure that the SD card is not reformatted.

## 2.2 Possible fields of application of the Inline controller

### 2.2.1 ILC 130 ETH

The **ILC 130 ETH** Inline controller can be used as a distributed control system of an Inline station, which is connected to an Ethernet system. An Inline local bus (Figure 2-1) can then be connected to the Inline controller.

**Inline local bus**

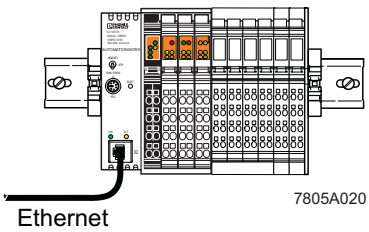



Figure 2-1 Connected Inline local bus

Please note that the ILC 130 ETH does not support connection of the INTERBUS remote bus.

### 2.2.2 ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, and ILC 190 ETH 2TX

The Inline controllers (ILC 150 ETH/ILC 155 ETH/ILC 170 ETH 2TX/ILC 190 ETH 2TX) can be used as a distributed control system of an Inline station, which is connected to an Ethernet system. A single Inline local bus (Figure 2-2) as well as a complete INTERBUS system with a maximum of 4 remote bus levels (Figure 2-3) can be connected to the Inline controller.

**Inline local bus**

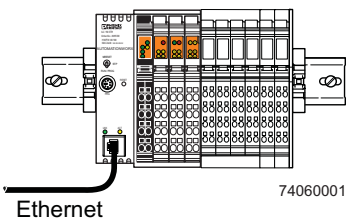



Figure 2-2 Connected Inline local bus

## Remote bus levels

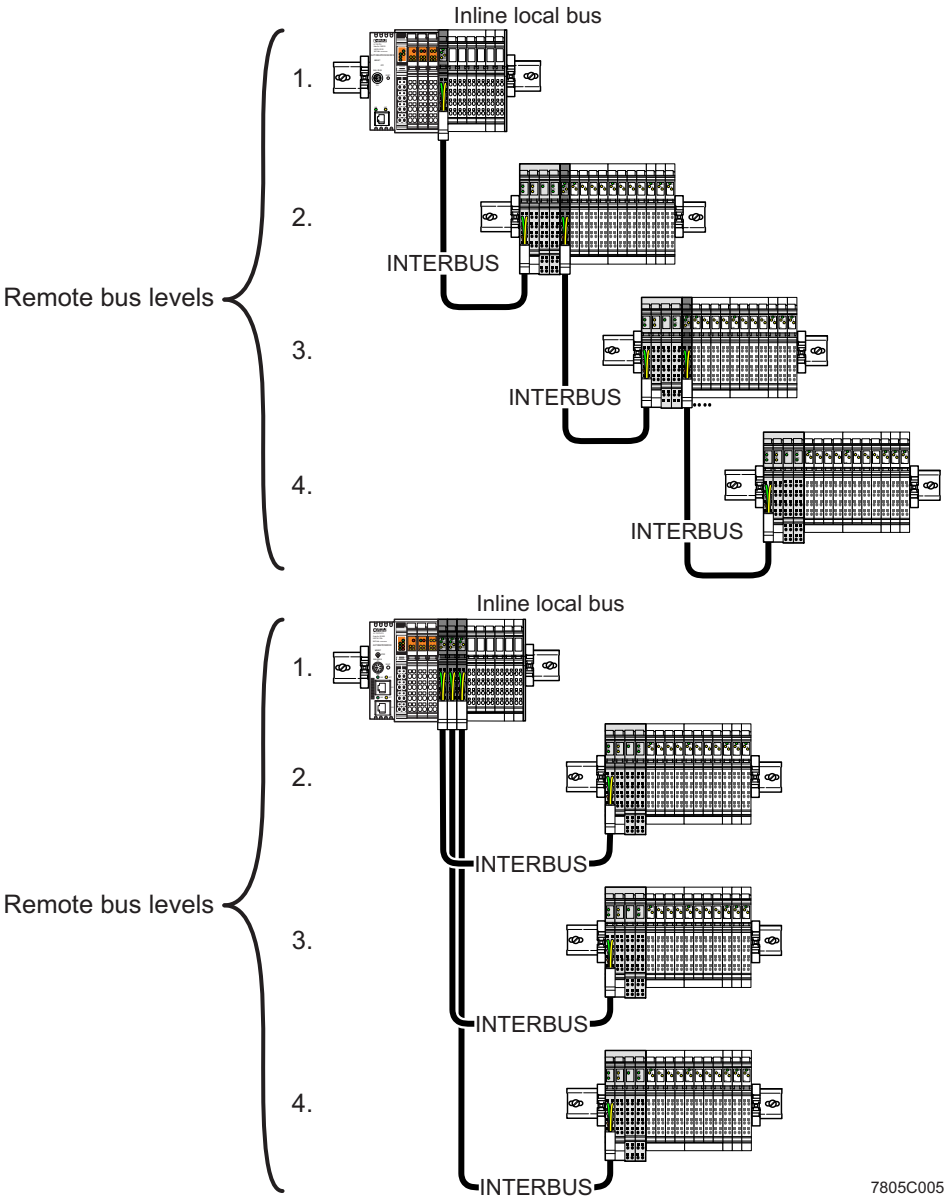



Figure 2-3 Remote bus levels

### 2.2.3 ILC 170 ETH 2TX and ILC 190 ETH 2TX as a PROFINET IO device in a PROFINET network

The following figure shows the example of a ILC 170 ETH 2TX as a PROFINET IO device in a PROFINET network.

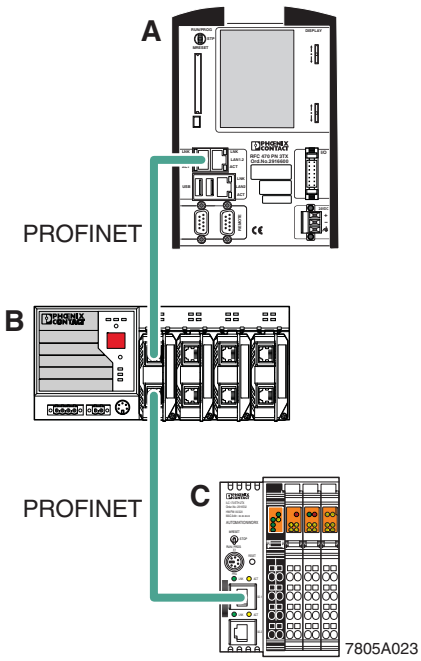



Figure 2-4 PROFINET IO device using the example of the ILC 170 ETH 2TX

Key:

A PROFINET IO controller (in the example RFC 470 PN 3TX Remote Field Controller)

B Managed switch

C PROFINET IO device (in the example: ILC 170 ETH 2TX)

For additional information on how to integrate the ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers as a PROFINET IO device in a PROFINET network, please refer to Section “ILC 170 ETH 2TX (ILC 190 ETH 2TX) as a PROFINET IO device” on page 3-5.

### 2.2.4 Applicative system redundancy with ILC 170 ETH 2TX or ILC 190 ETH 2TX

The following figure shows an example of applicative system redundancy. The example shows an IO device with control function (ILC 170 ETH 2TX) that is connected with a switch to a PROFINET network. Identical application programs run on both higher-level IO controllers. To achieve synchronization both IO controllers are connected with a switch over an Ethernet connection. IO controller A functions as a primary IO controller, IO controller B functions as a backup IO controller.

The ILC 190 ETH 2TX also supports applicative system redundancy.

Controllers with applicative (programmed) redundancy

Network redundancy with Media Redundancy Protocol (MRP)

IO devices with Phoenix Redundancy Layer

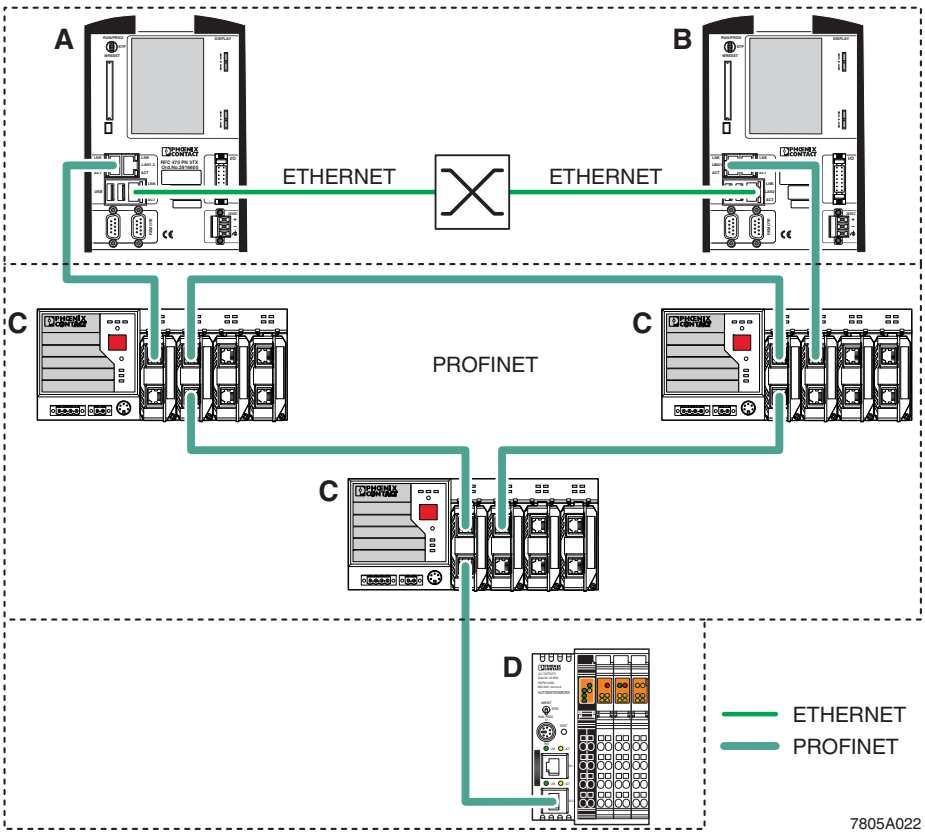



Figure 2-5 Applicative system redundancy – An example

For further information on applicative system redundancy, please refer to the AH EN APPLICATIVE SYSTEM REDUNDANCY. This application note can be downloaded at www.phoenixcontact.net/catalog.

A Primary IO controller (in the example: RFC 470 PN 3TX Remote Field Controller)

B Backup IO controller (in the example: RFC 470 PN 3TX Remote Field Controller)

C Managed switches

D IO device with control function
(in the example: ILC 170 ETH 2TX with IO device function and Phoenix Redundancy Layer)

## 2.3 Notes on using the Inline controller (ILC 150 ETH/ILC 155 ETH) in potentially explosive areas

**Approval according to EC directive 94/9 (ATEX) II 3G Ex nAC II T4 X**

The Inline controller (ILC 150 ETH/ILC 155 ETH) conforms to the requirements of protection type "n" and can be installed in a zone 2 potentially explosive area. The Inline controller (ILC 150 ETH/ILC 155 ETH) is a category 3G item of electrical equipment.

**WARNING: Explosion hazard**
**Only Inline terminals that are approved for use in potentially explosive areas may be snapped onto the Inline controller.**

Before using an Inline terminal in a zone 2 potentially explosive area, first check that the terminal has been approved for installation in this area.

For a list of terminals approved for zone 2 potentially explosive areas, please refer to the AH EN IL EX ZONE 2 application note.

Check the labeling on the Inline terminal and the packaging (see Figure 2-6).

Figure 2-6 Typical labeling of terminals for use in potentially explosive areas

**WARNING: Explosion hazard**

**Prior to startup, make sure that the following points and instructions are observed.**

1. When working on the Inline controller, always disconnect the supply voltage.
2. The Inline controller must only be installed, started up, and maintained by qualified specialist personnel.
3. Mount the Inline controller in a control cabinet or metal housing. The minimum requirement for both items is IP54 protection according to EN 60529.
4. The Inline controller must not be subject to mechanical strain and thermal loads, which exceed the limits specified in the product documentation.
5. The Inline controller must not be repaired by the user. Repairs may only be carried out by the manufacturer. The Inline controller is to be replaced by an approved controller of the same type.
6. Only category 3G equipment may be connected to Inline controllers in zone 2.
7. Observe all applicable standards and national safety and accident prevention regulations for installing and operating equipment.

**Restrictions**

**WARNING:** Explosion hazard

When using the controller in potentially explosive areas, observe the technical data and limit values specified in the corresponding documentation (user manual, package slip).

**WARNING:** Explosion hazard, restrictions regarding the Inline system

– Please make sure that the **maximum permissible current of 4 A** flowing through potential jumpers $U_M$ and $U_S$ (total current) is not exceeded when using the Inline terminals in potentially explosive areas.

## 2.4 Unpacking the Inline controller

The Inline controller is supplied in an ESD bag together with a package slip with installation instructions. Please read the complete package slip carefully before unpacking the Inline controller.

**NOTE: Electrostatic discharge**

The Inline controller contains components, which may be damaged or destroyed by electrostatic discharge. When handling the Inline controller, observe the necessary safety precautions against electrostatic discharge (ESD) according to EN 61340-5-1.

**NOTE:**

To avoid possible damage to the Inline controller, unpack and pack the controller in accordance with the ESD regulations.

## 2.5 Connection and operating elements

**ILC 130 ETH, ILC 150 ETH, ILC 155 ETH**

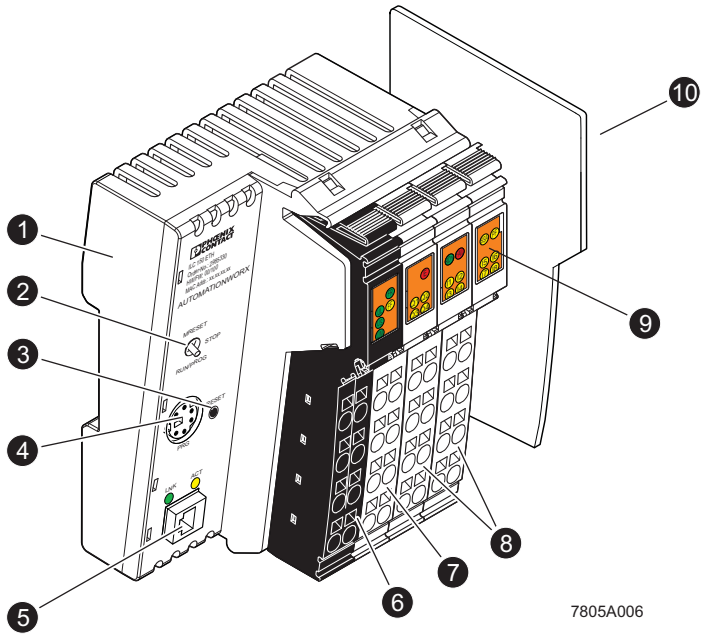



Figure 2-7 Structure of the Inline controller (ILC 130 ETH, ILC 150 ETH, ILC 155 ETH; shown in the figure: ILC 150 ETH)

The Inline controller consists of the following components:

**1** Electronics base
**2** Mode selector switch
**3** Reset button
**4** V.24 (RS-232) interface
**5** Ethernet connection
**6** Connector 1: terminal points for voltage supply
**7** Connector 2: output terminal points
**8** Connectors 3 and 4: input terminal points
**9** Diagnostic and status indicators
**10** End plate

**ILC 170 ETH 2TX,**
**ILC 190 ETH 2TX**

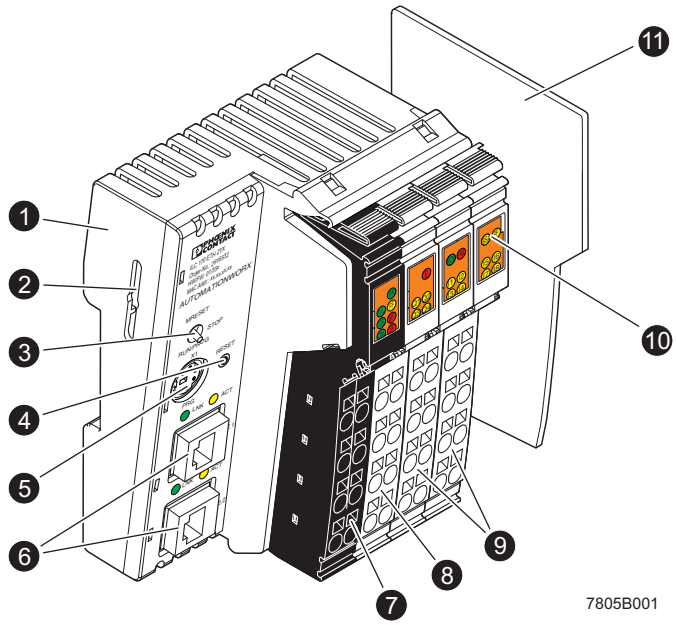



Figure 2-8 Structure of the Inline controller (ILC 170 ETH 2TX, ILC 190 ETH 2TX; shown in the figure: ILC 170 ETH 2TX)

The Inline controller consists of the following components:

**1** Electronics base

**2** Slot for the parameterization memory/card holder (SD card).

The SD card is not supplied as standard with the ILC 170 ETH 2TX/ILC 190 ETH 2TX Inline controllers.

Please refer to the ordering data in Section “Accessories” on page 5-8.

**3** Mode selector switch

**4** Reset button

**5** V.24 (RS-232) interface (X1)

**6** Ethernet interfaces (X2.1/X2.2)

**7** Connector 1: terminal points for voltage supply

**8** Connector 2: output terminal points

**9** Connectors 3 and 4: input terminal points

**10** Diagnostic and status indicators

**11** End plate

## 2.6 Diagnostic and status indicators

The descriptions for diagnostic and status indicators apply to all the Inline controllers listed on the inner cover page of this manual.

The diagnostic and status indicators are used for quick local error diagnostics.

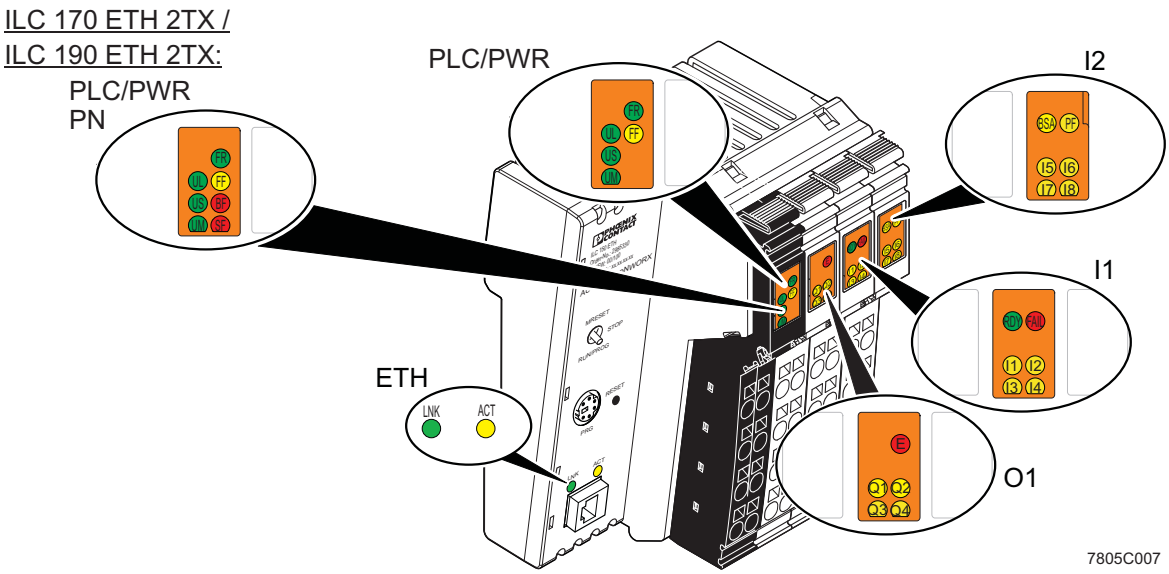



Figure 2-9 Diagnostic and status indicators

**Local diagnostic and status indicators**

| Des. | Color | Status | Meaning |
|---|---|---|---|
| **ETH: State of the Ethernet interface (applies to both interfaces (X2.1, X2.2) for the ILC 170 ETH 2TX/ILC 190 ETH 2TX)** | | | |
| **LNK** | Green | OFF | Connection not established successfully |
| | | ON | Connection established successfully (link): The Inline controller is able to contact another network device. |
| **ACT** | Yellow | OFF | Data transmission inactive |
| | | ON | Data transmission (activity): The Ethernet interface transmits or receives data. |
| **PROFINET: (ILC 170 ETH 2TX/ILC 190 ETH 2TX)** | | | |
| **BF** | Red | | Status of PROFINET IO communication/communication error (BusFail) |
| | | OFF | The PROFINET IO controller has established an active communication connection to at least one PROFINET IO device. |
| | | ON | No PROFINET IO communication available (no link status at the Ethernet ports). |
| | | Flashing | No communication connection to the PROFINET IO controller, the connection is being established. Link status available. |
| **SF** | Red | | System failure (PROFINET) |
| | | OFF | PROFINET diagnostics not present. |
| | | ON | PROFINET diagnostics present. |

| Des. | Color | Status | Meaning |
|---|---|---|---|
| **PLC: Diagnostics of the Inline controller** | | | |
| **FR** | Green | | Inline controller running |
| | | OFF | IEC 61131 runtime system not ready to operate. |
| | | Flashing | IEC 61131 runtime system successfully initialized.<br>Control function in READY/STOP state, program not processed. |
| | | ON | IEC 61131 runtime system successfully initialized and a program is running.<br>Control function in RUN state. |
| **FF** | Yellow | | Failure |
| | | ON | A runtime error has occurred in the IEC 61131 runtime system program |
| | | OFF | No runtime error has occurred in the IEC 61131 runtime system program |
| **FR + FF** | | Flashing | Applies to ILC 170 ETH 2TX and ILC 190 ETH 2TX only:<br>There is no SD card in the Inline controller. |
| **PWR: Supply voltage**<br>**(See also Section "Remote bus" on page 2-25)** | | | |
| **UL** | Green | | 24 V supply $U_{ILC}$ for generation of the voltages $U_L$ and $U_{ANA}$ |
| | | OFF | Supply voltage not present |
| | | ON | Supply voltage present<br>(indication if 24 V supply voltage $U_{ILC}$ present) |
| **US** | Green | | 24 V supply for segment circuit |
| | | OFF | Supply voltage not present |
| | | ON | Supply voltage present |
| **UM** | Green | | 24 V supply for main circuit |
| | | OFF | Supply voltage not present |
| | | ON | Supply voltage present |
| **IL: INTERBUS diagnostics** | | | |
| **RDY** | Green | | INTERBUS master ready to operate/data transmission active<br>(INTERBUS ready/running) |
| | | OFF | INTERBUS master not ready to operate. |
| | | Flashing | INTERBUS master in READY or ACTIVE state. |
| | | ON | INTERBUS master in RUN state. |
| **FAIL** | Red | | Failure |
| | | OFF | No error occurred |
| | | ON | One of the following errors has occurred:<br>– Bus error in the connected bus (remote bus/local bus)<br>– Controller error |
| **BSA** | Yellow | | Bus segment aborted |
| | | OFF | Bus segment(s) in the connected bus not switched off. |
| | | ON | One or more bus segments in the connected bus are switched off. |

| Des. | Color | Status | Meaning |
|---|---|---|---|
| **PF** | Yellow | | Peripheral fault |
| | | OFF | Peripheral fault on a device in the connected bus |
| | | ON | Peripheral fault of a device in the connected bus (local bus or remote bus) |
| **I/O: Digital inputs and outputs** | | | |
| **I1 to I8** | Yellow | | Inputs 1 to 8 |
| | | OFF | Corresponding input not set |
| | | ON | Corresponding input is set |
| **E** | Yellow | | Error |
| | | OFF | No short circuit/overload at one of the outputs 1 to 4 |
| | | ON | Short circuit/overload at one of the outputs 1 to 4 |
| **Q1 to Q4** | Yellow | | Outputs 1 to 4 |
| | | OFF | Corresponding output not set |
| | | ON | Corresponding output is set |

## 2.7 Mode selector switch

The mode selector switch is used to define the operating state of the application program.

The RUN/PROG and STOP positions have a toggle button function and the MRESET position a pushbutton function. After releasing the button in the MRESET position, it returns to the STOP position.

Table 2-1 Inline controller operating modes

| Operating mode | Explanation |
|---|---|
| RUN/PROG | The application program is in the RUN state.<br>The PC Worx/PC Worx Express software can be used for program and configuration modifications.<br>The monitoring and online functions can be used. |
| STOP | The application program is in the STOP state. |
| MRESET | Retain data and the application program are deleted.<br>Set the mode selector switch in the following sequence to perform deletion of retain data and the application program:<br>• Set the switch to the MRESET position for three seconds.<br>• Release the switch for less than three seconds.<br>• Set the switch to the MRESET position for three seconds. |

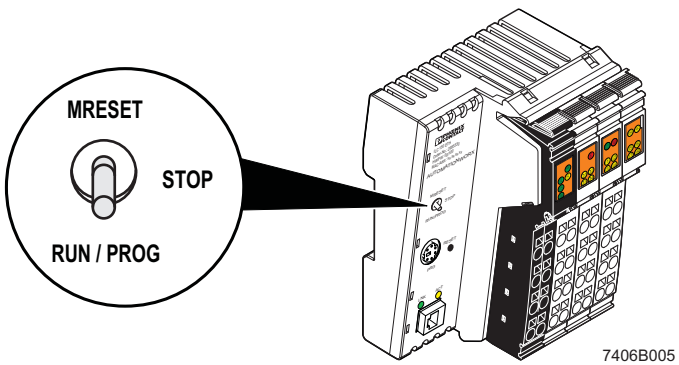



Figure 2-10 Mode selector switch

## 2.8 Reset button (concealed)

The reset button (see item 3 in Figure 2-7 on page 2-9 or item 4 in Figure 2-8 on page 2-10) on the Inline controller can only be operated with a pin and is therefore protected against accidental activation.

If you carry out a voltage reset at the same time as you press the reset button, this resets the Inline controller to its default settings.

**Procedure**

Hold down the reset button and switch the supply voltage of the Inline controller off and on again. Release the reset button only after the FR (green) and FF (yellow) LEDs flash alternately.

The Inline controller has been initialized successfully and reset to its default settings only after the FR (green) and RDY (green) LEDs are flashing. The control function is in the READY/STOP state, a program is not processed. This procedure can take up to a minute, approximately.

## 2.9 Parameterization memory

The parameterization memory can be used to save programs and configurations, which belong to your project. In addition, application-specific data can also be stored on the parameterization memory. See also Section “Parameterization memory and Internet Explorer” on page 3-12.

**ILC 130 ETH, ILC 150 ETH, ILC 155 ETH**

The ILC 130 ETH, ILC 150 ETH, and ILC 155 ETH Inline controllers have an integrated parameterization memory.

**ILC 170 ETH 2TX, ILC 190 ETH 2TX**

The ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers have a plug-in parameterization memory in the form of an SD card. This plug-in parameterization memory is essential for operating the Inline controller. The parameterization memory is available in the following size.

**NOTE:** It is not permitted to operate the ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers without parameterization memory.

The parameterization memory is recognized during initialization of the Inline controller. Ensure that the parameterization memory has been inserted before switching on the controller to enable the controller to use it.

Only insert and remove the parameterization memory when the Inline controller supply voltage is disconnected.

The parameterization memory must not be removed during operation of the Inline controller.

Use only a parameterization memory provided by Phoenix Contact (for the ordering data, see Section “Accessories” on page 5-8).

**NOTE: Parameterization memory (SD card) – formatting note**

The SD card is preformatted and designed for use with Phoenix Contact devices. Ensure that the SD card is not reformatted.

**Inserting/removing the parameterization memory**

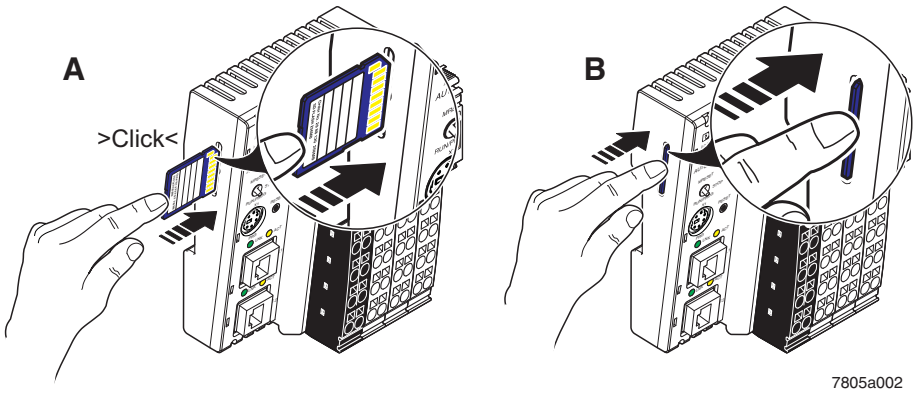



Figure 2-11 Inserting (A) and removing (B) the parameterization memory

**Inserting the SD card**

The ILC 170 ETH 2TX and ILC 190 ETH 2TX have an SD card holder with push/push technology.

- Insert the parameterization memory (SD card) into the slot as shown in Figure 2-11 (A).
- Applying light pressure, push the parameterization memory into the slot until it engages with a click in the card holder.

**Removing the SD card**

- Applying light pressure, push the parameterization memory into the slot in the direction shown in Figure 2-11 (B) until the snap-on mechanism releases the parameterization memory and partially ejects it from the slot. Remove the parameterization memory.

For additional information about the parameterization memory, please refer to Section “Parameterization memory and Internet Explorer” on page 3-12.

## 2.10 Internal basic circuit diagram

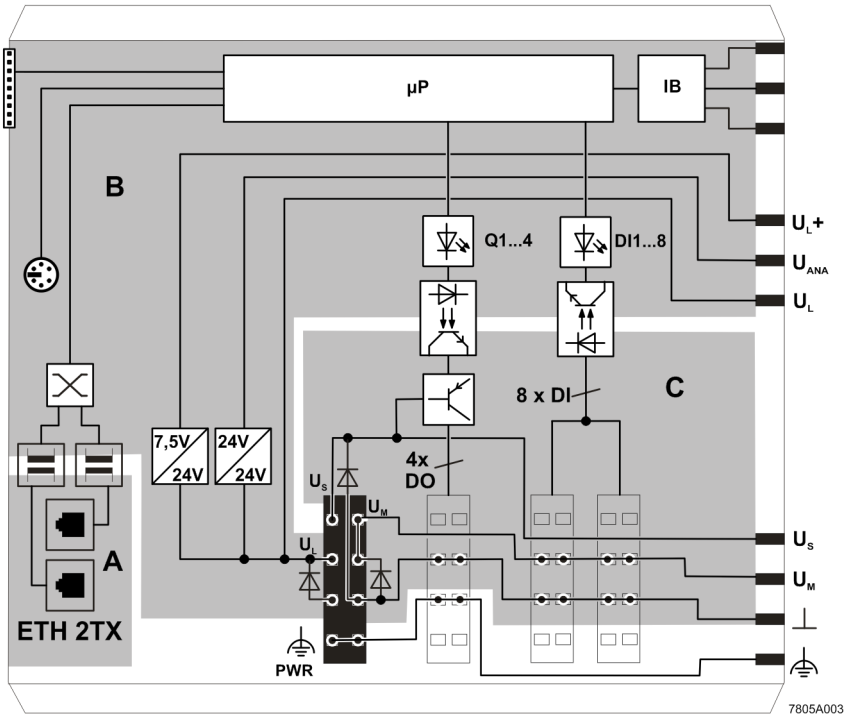



Figure 2-12 Internal basic circuit diagram (ILC 170 ETH 2TX)

Key:

| | | | |
|---|---|---|---|
| µP | Microprocessor | | Converter |
| IB | Protocol chip | | LED |
| | V.24 (RS-232) interface | | Optocoupler |
| | Transmitter | | NPN transistor |
| | RJ45 socket | | Ethernet switch |
| | SD card holder (the SD card is not supplied as standard) | | |

The gray areas in the basic circuit diagram represent electrically isolated areas.:

A: Ethernet interface
B: Logic
C: I/O

Other symbols used are explained in the IL SYS INST UM E user manual.

## 2.11 Mounting and removing the Inline controller

For notes and instructions on mounting and removing Inline terminals, please refer to the IB IL SYS PRO UM E user manual (for INTERBUS), the IL SYS INST UM E Inline installation manual or the Inline system manual for your bus system.

**NOTE:**

Before mounting or removing the Inline controller, ensure that the power supply is switched off and cannot be switched on again by unauthorized persons.

An Inline station is set up by mounting the individual components side by side. No tools are required. Mounting the components side by side automatically creates potential and bus signal connections between the individual station components.

The controller is mounted perpendicular to the DIN rail.

**Mounting location**

Like all other terminals of the Inline product range, the Inline controller has IP20 protection and is designed for use in a closed control cabinet or control box (terminal box) with IP54 protection or higher.

**DIN rail**

The Inline controller is mounted on a 35 mm standard DIN rail.

Fix the DIN rail, on which the Inline controller is mounted several times, especially in the area around the Inline controller. This makes it easier to remove the Inline controller.

**End clamp**

Mount end clamps on both sides of the Inline station. The end clamps ensure that the Inline station is correctly mounted. End clamps fix the Inline station on both sides and keep it from moving from side to side on the DIN rail. Phoenix Contact recommends using CLIPFIX 35-5 end clamps (Order No. 3022276).

**End plate**

The mechanical end of an Inline station is the end plate. It has no electrical function. It protects the station against ESD pulses and the user against dangerous contact voltages. The end plate is supplied together with the Inline controller and does not need to be ordered separately.

**NOTE:**

When mounting or removing the Inline controller it must be tilted. This means that no Inline terminal should be installed directly to the right of the Inline controller during mounting and removal. The terminal must be removed prior to mounting or removing the Inline controller. Otherwise, the locking clips and jumper contacts will be damaged.

**Mounting position**

Mount the Inline controller horizontally (as shown in Figure 2-13 on page 2-19). The specified temperature range (see “Ambient conditions” on page 5-6) is only guaranteed if the Inline controller is mounted in this position.

**Mounting**

When mounting the Inline controller, proceed as shown in Figure 2-13:

- Disconnect the power to the station.
- Place the Inline controller onto the DIN rail from above (A) and push down (B).
- Then attach all the electronics bases required to set up the station. Observe the information provided in the above user manuals.

Ensure that **all** featherkeys and keyways on adjacent terminals are securely interlocked.

- Once all the bases have been snapped on, plug the connectors into the appropriate bases.
  First, place the front connector shaft latching in the front snap-on mechanism (C1).
  Then press the top of the connector towards the base until it snaps into the back snap-on mechanism (C2).

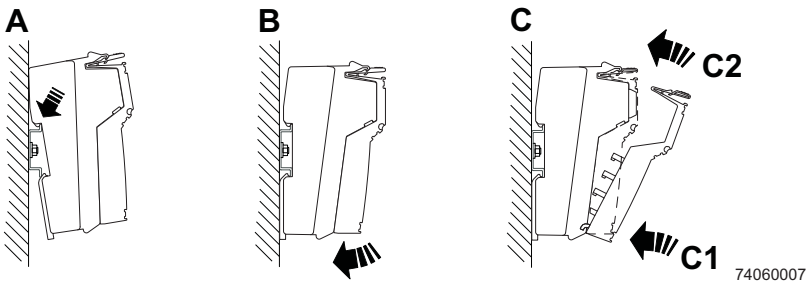



Figure 2-13 Snapping on the Inline controller

**Removal**

When removing the Inline controller, proceed as shown in Figure 2-15 on page 2-20:

- Disconnect the power to the station.

Unlike other Inline terminals, the Inline controller is removed by tilting it away from the DIN rail. This requires the Inline terminal to the right to be removed prior to removing the Inline controller. The right connector of the Inline controller must also be removed.

Remove the third and fourth connectors to access the right base latch.

It is therefore recommended that all connectors be removed prior to removing the Inline controller.

- If the connectors cause trouble during removal
  Remove all the connectors of the Inline controller.
  - Lever up each connector by pressing on the back connector shaft latching (Figure 2-15, A1).
  - Remove the connectors (Figure 2-15, A2).
- If Inline terminals are installed next to the Inline controller (see Figure 2-14):
  - Remove the following connectors:
    – All connectors of the terminal directly connected (A1 to A4)
    – The adjacent connector of the following terminal (B1)
  - Remove the directly adjacent Inline terminal (A).

Observe the information provided in the above user manuals.

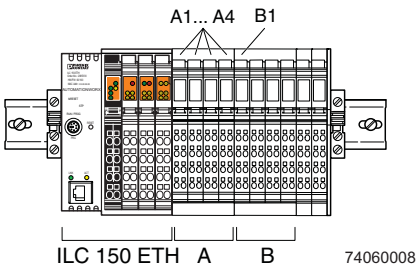



Figure 2-14 Connectors to be removed if terminals are installed next to the Inline controller

- Insert a tool in the base latches of the Inline controller and pull gently upwards (Figure 2-15, B). Pull out the Inline controller from the DIN rail (C1, C2).

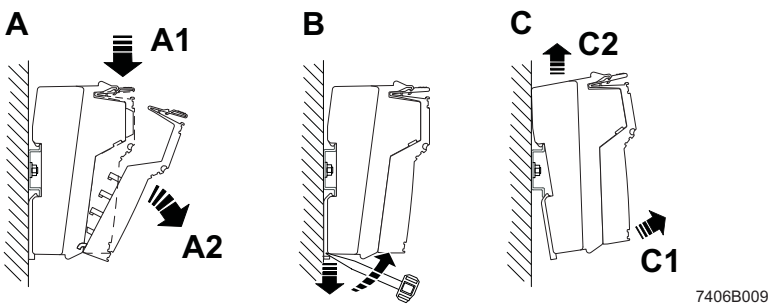



Figure 2-15 Removing the Inline controller

**Replacing the Inline controller**

If you want to replace an Inline controller within an Inline station, proceed as described above (removing and mounting). Make sure that the terminal to the right is not installed when removing and mounting the Inline controller. Only reinstall this terminal once the Inline controller is mounted.

In particular, make sure that **all** featherkeys and keyways on adjacent terminals are securely interlocked.

Observe the following when replacing an Inline controller:
Enter the new MAC address when using the BootP server.

## 2.12 Communication paths

The communication path to the Inline controller must be determined before communication with the Inline controller can take place.

The following communication paths are available on the Inline controller:

| | | | |
|---|---|---|---|
| **ILC 130 ETH**, ILC 150 ETH, ILC 155 ETH: | (A1) | 1 x Ethernet | 10/100 BASE-T(X) |
| **ILC 170 ETH 2TX**, ILC 190 ETH 2TX: | (A2) | 2 x Ethernet | X2.1/X2.2: 10/100 BASE-T(X) (switched internally) |
| **ILC 130 ETH**, ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX: | (B) | PRG | The serial interface of your PC is directly connected to the Inline controller (not to the programming).<br>For additional information about using the serial interface (e.g., IP address assignment), please refer to Section “Serial PRG interface - function blocks” on page 2-24. |

The Inline controller cannot be programmed via the V.24 (RS-232) PRG interface.

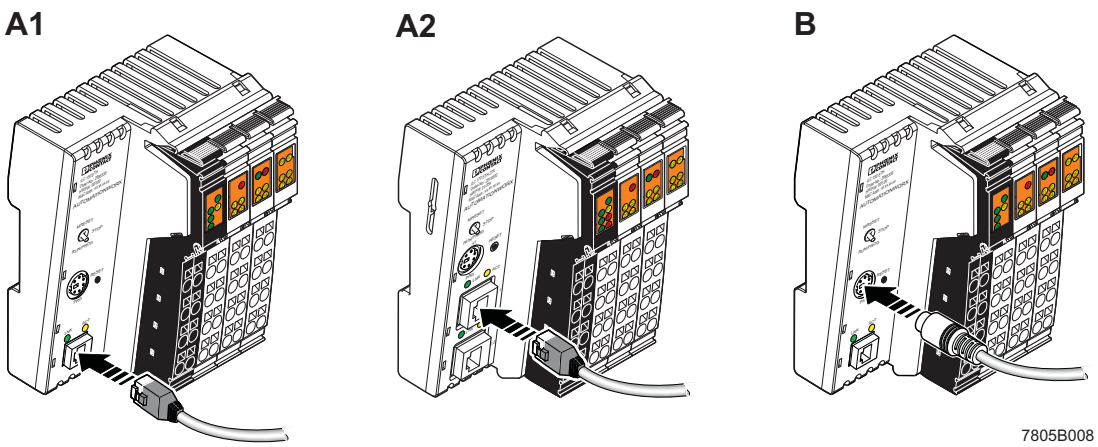



Figure 2-16 Communication paths: (A1/A2) Ethernet (B) PRG

### 2.12.1 Ethernet

A standardized Ethernet interface each is available for connecting the ILC 130 ETH, ILC 150 ETH, and ILC 155 ETH Inline controllers to the Ethernet network. For the ILC 170 ETH 2TX and ILC 190 ETH 2TX Inline controllers, two standardized Ethernet interfaces (X2.1/X2.2) are available for connection to the Ethernet network.

The Ethernet network is connected via RJ45 female connectors.

Use at least an Ethernet cable according to Cat. 5 of IEEE 802.3.
Observe the bending radii of the Ethernet cables used.

The contact assignment of the interface is as follows:

| | | |
|---|---|---|
| Transmit data + | T + | 1 |
| Transmit data - | T - | 2 |
| Receive data + | R + | 3 |
| - | | 4 |
| - | | 5 |
| Receive data - | R - | 6 |
| - | | 7 |
| - | | 8 |

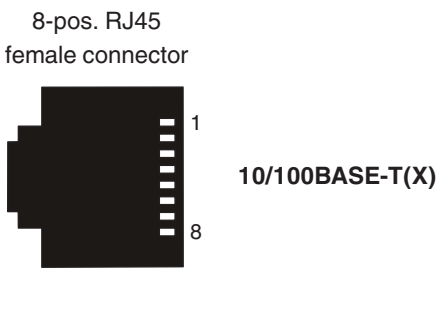



Figure 2-17 Ethernet interface

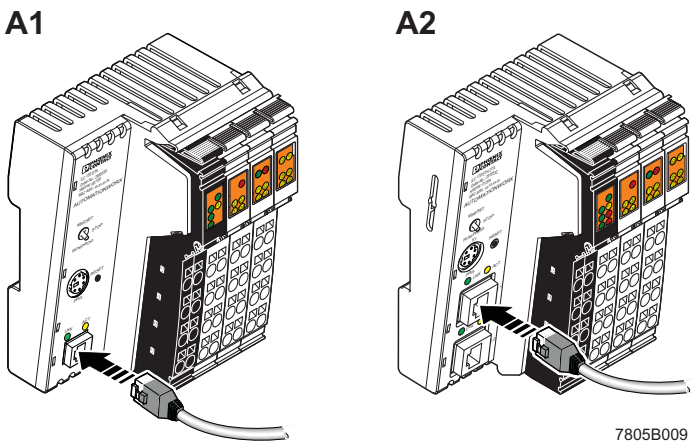



Figure 2-18 Connecting the Ethernet cable to the Inline controller
A1: ILC 130 ETH, ILC 150 ETH, ILC 155 ETH
A2: ILC 170 ETH 2TX, ILC 190 ETH 2TX

The interface is able to switch-over the transmitter and receiver automatically (auto crossover).

## 2.12.2 Serial PRG interface (mini-DIN female connector)

This serial interface providing the Ethernet interface enables communication with the Inline controller from a PC.

A connecting cable is required for connecting the Inline controller to a PC with PC Worx/PC Worx Express via the serial PRG interface.

A connecting cable is required for direct connection of the Inline controller to a PC with PC Worx. Connect the connecting cable to the programming interface of the Inline Controller (designation "PRG") and the serial interface of the PC.

This interface can be used to **either** assign the IP address of the Inline controller and access the Inline controller using the Diag+ diagnostic tool **or** communicate with special I/O devices via function blocks (see Section 2.12.3, “Serial PRG interface - function blocks”).

The Inline controller cannot be programmed via the V.24 (RS-232) interface.

**Connection**

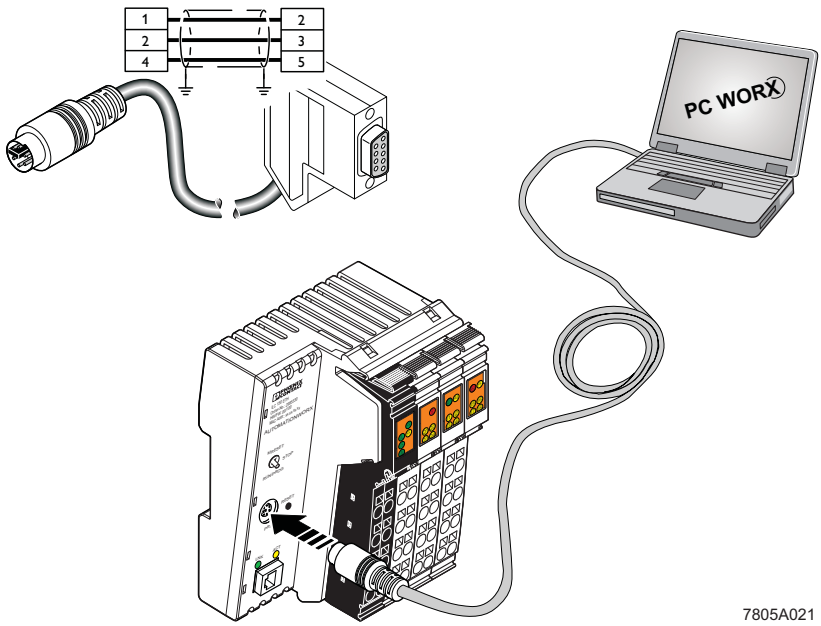



Figure 2-19 Connecting cable between PC and Inline controller

Ordering data:
Connecting cable for connecting the Inline controller to a PC (V.24 (RS-232))
for PC Worx, length 3 m (Designation PRG CAB MINI DIN, Order No. 2730611).

### 2.12.3 Serial PRG interface - function blocks

This interface can be used for the following:

– Assignment of the IP address or working with Diag+
– Communication with I/O devices via function blocks

The following function blocks are available in the PC Worx/PC Worx Express software:

Table 2-2 Function block overview

| Function block | Short description |
|---|---|
| RS232_INIT | Parameterization of the serial interface<br><br>You can use this function block to specify the following parameters of the serial interface:<br><br>– Protocol: Transparent<br>– Baud rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600 or 115200<br>– Data width: 8 data bits, even parity<br>– Number of stop bits: 1<br>– Hardware flow control: None |
| RS232_RECEIVE | Reading the internal receive memory of the serial interface |
| RS232_SEND | Data transmission to the internal transmit memory of the serial interface |

Once the RS232_INIT function block has been activated, the interface is parameterized accordingly so that it is only possible to communicate with the connected I/O devices.

Deactivating the RS232_INIT function block or performing another cold restart or warm start enables the IP address to be assigned and the controller to be accessed with Diag+.

For additional information about the function blocks, please refer to the online help for the PC Worx/PC Worx Express software.

#### I/O devices that can be connected

Various I/O devices (e.g.,modem, printer, barcode scanner) can be connected to the controller via the serial PRG interface. The connected devices are addressed from the application program using function blocks.

## 2.13 INTERBUS

The descriptions for INTERBUS apply to all Inline controllers listed on the inner cover page of this manual.

Observe the information in the "Configuring and installing the Inline product range" user manual IB IL SYS PRO UM E when creating an Inline system (local bus and/or remote bus).

Please note that the Inline controller does not support the following functions:

- Switching of devices
- Single-channel diagnostics
- Fiber optic diagnostics/optical regulation
- Logical addressing

Only INTERBUS devices with SUPI 3 and SUPI 3 OPC protocol chip can be used with INTERBUS as local bus/remote bus devices.

### 2.13.1 Local bus

The local bus is automatically created by directly connecting I/O modules to the Inline controller.

### 2.13.2 Remote bus

Please note that the ILC 130 ETH does not support connection of the INTERBUS remote bus.

Connect the remote bus to the Inline controller using one of the following branch terminals. They only differ in the scope of supply.

IBS IL 24 RB-T (Order No. 2727941)

IBS IL 24 RB-T-PAC (Order No. 2861441; including accessories)

The **first branch terminal** must be placed directly after the Inline controller. In terms of topology, it opens a **remote bus**.

If **additional branch terminals** are used after the first branch terminal, they must be installed directly one after the other (see also notes in the terminal-specific data sheet). In terms of topology, the additional branches are **remote bus branches** with the branch terminal being the first device in the corresponding remote bus branch.

A maximum of 3 branch terminals can be connected to the Inline controller, each of which opens a remote bus (see Figure 2-3 on page 2-4).

## 2.14 Power supply

The descriptions for the power supply apply to all the Inline controllers listed on the inner cover page of this manual.

### 2.14.1 Sizing of the power supply

Use a power supply unit suitable for the currents in your application. The selection depends on the bus configuration, the resulting maximum currents, and the type of supply (separate supply of $U_{ILC}$, $U_M$, and $U_S$, or supply from a power supply unit).

A power supply without a fall-back characteristic curve must be used for correct operation of the Inline controller (see Figure 2-21).
When the Inline controller is switched on, an increased inrush current is temporarily triggered. The Inline controller behaves like a capacitive load when it is switched on.

Some electronically controlled power supplies have a fall-back characteristic curve (see Figure 2-20). They are not suitable for operation with capacitive loads.

A primary switched-mode power supply unit (without fall-back characteristic curve) from the QUINT POWER range (see INTERFACE catalog from Phoenix Contact) is recommended for Inline controller operation.

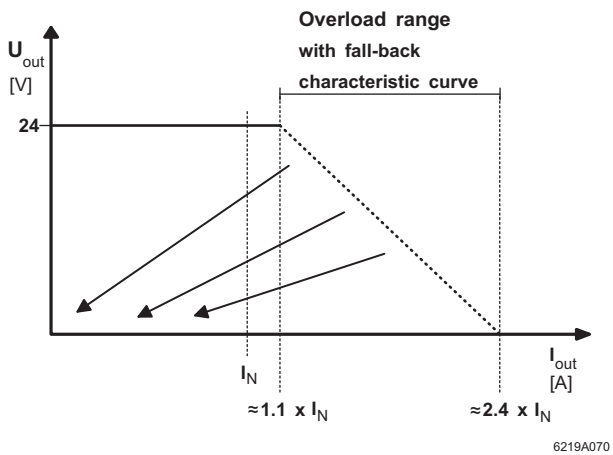



Figure 2-20 Overload range **with** fall-back characteristic curve

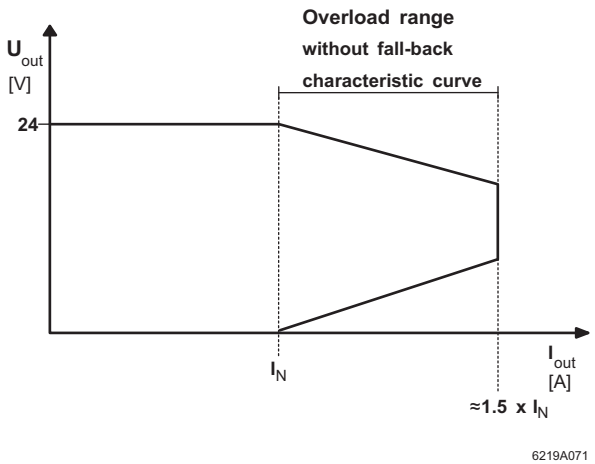



Figure 2-21 Overload range **without** fall-back characteristic curve

## 2.14.2 Power supply connection

Supply the Inline controller using external 24 V DC voltage sources. The permissible voltage ranges from 19.2 V DC to 30 V DC (ripple included).
The power consumption of the Controller at 24 V is typically 4.8 W (no local bus devices connected).

Only use power supplies that are suitable for operation with capacitive loads (increased inrush current) (see Section “Sizing of the power supply” on page 2-26).

1. Connect the power supplies to the connector for power supply as shown in Figure 2-22.
2. Insert the connector in the Inline controller.
3. Switch on the power supplies.
4. The UL, UM, and US LEDs light up and, after around 10 seconds, the FR and RDY LEDs start flashing.

The Inline controller is now fully initialized.

If the LEDs do not light up or start flashing, there is a serious fault on the Inline controller. In this case, please contact Phoenix Contact.

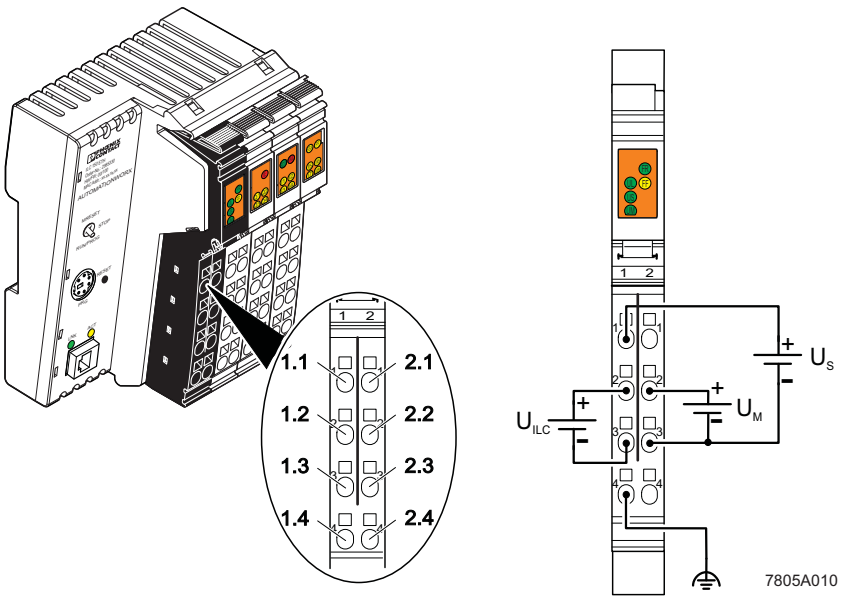



Figure 2-22 Supply voltage connection

| Terminal point | Assignment | | Notes |
|---|---|---|---|
| **Connector 1** | **Power connector** | | |
| **1.1** | 24 V DC ($U_S$) | 24 V segment voltage supply | The supplied voltage is directly routed to the potential jumper.<br>**NOTE:**<br>Protect the supply voltage externally according to the connected load (local bus devices) with **8 A, maximum**. Make sure the external fuse blows in any case. |
| **1.2** | 24 V DC ($U_{ILC}$) | 24 V supply | The 7 V communications power ($U_L$) for the ILC and the connected local bus devices is generated from this voltage. The 24 V analog power ($U_{ANA}$) for the local bus devices is also generated.<br>**NOTE:**<br>Protect the supply voltage externally according to the connected load (local bus devices) with **2 A, maximum**. Make sure the external fuse blows in any case. |
| **2.1, 2.2** | 24 V DC ($U_M$) | 24 V main voltage supply | The main voltage is routed to the local bus devices via the potential jumpers.<br>**NOTE:**<br>Protect the supply voltage externally according to the connected load (local bus devices) with **8 A, maximum**. Make sure the external fuse blows in any case. |
| **1.3** | LGND | Reference potential logic ground | The potential is reference ground for the communications power. |
| **2.3** | SGND | Reference potential segment ground | The reference potential is directly led to the potential jumper and is, at the same time, reference ground for the main and segment supply. |
| **1.4, 2.4** | FE | Functional earth ground (FE) | Functional earth ground must be connected through the power supply. The contacts are directly connected to the potential jumper and FE springs on the bottom of the housing. The Inline controller is grounded when it is snapped onto a grounded DIN rail. Functional earth ground is only used to discharge interference. |

**NOTE:**

– The **maximum total current** flowing through the potential jumpers is **8 A**.

### 2.14.3 24 V segment supply/24 V main supply

The segment supply and main supply must have the same reference potential. An electrically isolated voltage area is not possible.

### 2.14.4 24 V segment supply

There are several ways of providing the segment voltage at connector 1:

1. You can provide the segment voltage separately at the terminal points 1.1 and 2.3 (GND) (see Figure 2-22 on page 2-27).
2. You can jumper connections 1.1 and 2.1 (or 2.2) to ensure that the segment circuit is supplied from the main circuit.
3. You can create a switched segment circuit with a switch between terminal points 1.1 and 2.1 (or 2.2).

**NOTE:**

The 24 V segment supply has protection against polarity reversal and surge voltage.

It does not have short-circuit protection.

The user must provide short-circuit protection. The rating of the preconnected fuse must be such that the maximum permissible load current of 8 A is not exceeded (total current at $U_M$ and $U_S$).

### 2.14.5 24 V main voltage

**NOTE:**

The 24 V main supply has protection against polarity reversal and surge voltage.

It does not have short-circuit protection.

The user must provide short-circuit protection. The rating of the preconnected fuse must be such that the maximum permissible load current of 8 A is not exceeded (total current at $U_M$ and $U_S$).

### 2.14.6 24 V ILC supply

**NOTE:**

The 24 V ILC supply has protection against polarity reversal and surge voltage. These protective elements are only used to protect the power supply unit.

The rating of the preconnected fuse must be such that the maximum permissible load current of 2 A is not exceeded.

### 2.14.7 Jumpers

Terminals 1.3 and 2.3 on connector 1 can be jumpered if the communications power and the segment power are not to be electrically isolated.

## 2.15 Digital inputs and outputs

The descriptions for digital inputs and outputs apply to all the Inline controllers listed on the inner cover page of this manual.

There are eight 24 V DC inputs and four 24 V DC outputs.

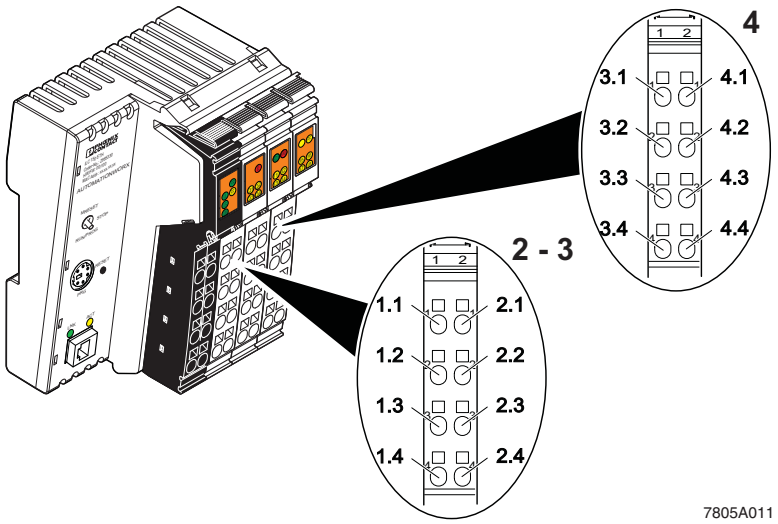



Figure 2-23 Assignment of the terminal points of connectors 2 to 4

Table 2-3 Terminal point assignment

| Terminal point | Assignment | Notes |
|---|---|---|
| **Connector 2** | **Output terminal points** | |
| **1.1** | Q1 | Output 1 |
| **2.1** | Q2 | Output 2 |
| **1.2, 2.2** | GND | Ground contact for 2 and 3-wire termination |
| **1.3, 2.3** | FE | Functional earth ground for 3-wire termination |
| **1.4** | Q3 | Output 3 |
| **2.4** | Q4 | Output 4 |
| The outputs are supplied with 24 V DC from the segment supply ($U_S$). | | |
| **Connector 3** | **Input terminal points** | |
| **1.1** | I1 | Input 1 |
| **2.1** | I2 | Input 2 |
| **1.2, 2.2** | 24 V | Supply voltage $U_M$ for 2 and 3-wire termination |
| **1.3, 2.3** | GND | Ground contact for 3-wire termination |
| **1.4** | I3 | Input 3 |
| **2.4** | I4 | Input 4 |

Table 2-3 Terminal point assignment (continued)

| Terminal point | Assignment | Notes |
|---|---|---|
| **Connector 4** | **Input terminal points** | |
| **3.1** | I5 | Input 5 |
| **4.1** | I6 | Input 6 |
| **3.2, 4.2** | 24 V | Supply voltage $U_M$ for 2 and 3-wire termination |
| **3.3, 4.3** | GND | Ground contact for 3-wire termination |
| **3.4** | I7 | Input 7 |
| **4.4** | I8 | Input 8 |

The inputs are supplied with 24 V DC from the main supply ($U_M$).

The outputs have protection when the ground connection is interrupted and must be wired accordingly.

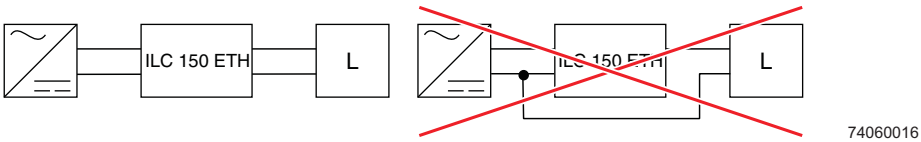



Figure 2-24 Basic wiring of an output with a load (L)
(shown using the ILC 150 ETH as an example)

Phoenix Contact recommends that connectors for digital 4-channel or 16-channel Inline terminals are used to connect sensors or actuators in 3-wire termination (not included in the scope of supply, see Section “Accessories” on page 5-8).

# 3 The Inline controller under PC Worx/PC Worx Express

## 3.1 Software version

Using the Inline controller requires the following PC Worx/PC Worx Express version or later:

Tabelle 3-1 Software version information

| Inline controller | Firmware version | | | | | |
|---|---|---|---|---|---|---|
| | ≤ 3.7x | | | ≥ 3.9x | | |
| | Software versions (Service Pack = SP, Hotfix = HF) | | | | | |
| | PC Worx | PC Worx Express | AUTOMATIONWORX Software Suite | PC Worx | PC Worx Express | AUTOMATIONWORX Software Suite |
| ILC 130 ETH | ≥ 5.20 SP 3 | | 2008 1.40 SP 3 | ≥ 6.1 SP 1 HF 1 | ≥ 6.1 SP 1 HF 1 | 2010 1.60 SP 1 |
| ILC 150 ETH | ≥ 5.10 SP 1 | – | 2007 1.30 SP 1 | | | |
| | – | ≥ 5.20 | 2008 1.40 | | | |
| ILC 155 ETH | ≥ 5.20 | | 2008 1.40 | | | |
| ILC 170 ETH 2TX | ≥ 5.20 SP 2 | | 2008 1.40 SP 2 | | | |
| ILC 190 ETH 2TX | ≥ 6.10 SP 1 | | 2010 1.60 SP 1 | | | |

**PROFINET IO device functions**

**ILC 170 ETH 2TX and ILC 190 ETH 2TX**

Please observe that the PROFINET IO device functions of these controllers are only available with the following software versions or later:

ILC 170 ETH 2TX PC Worx Version 6.00 Service Pack 2 or later (part of the AUTOMATIONWORX Software Suite 2009 1.50 Service Pack 2)

ILC 190 ETH 2TX PC Worx Version 6.10 Service Pack 1 or later (part of the AUTOMATIONWORX Software Suite 2010 1.60 Service Pack 1)

The PC Worx Express software does not support this function.

For information about installing and using PC Worx/PC Worx Express, please refer to the corresponding quick start guide. It can be downloaded at www.phoenixcontact.net/catalog and is supplied with the software.

## 3.2 Assigning the IP address for the controller/ BootP server

The procedure for assigning the IP address is essentially the same in PC Worx and PC Worx Express for all Inline controllers described in this manual. The following example describes the setting in PC Worx Express for the ILC 150 ETH.

By default upon delivery, the Inline controller has no preset IP address. Initial setting of the IP address can be carried out with the PC Worx/PC Worx Express software manually via the serial interface or by means of a BootP server. The IP address can be changed later via the serial connection or Ethernet using the PC Worx/PC Worx Express software.

The connecting cable (V.24 (RS-232) cable) PRG CAB MINI DIN (Order No. 2730611) is available as an optional accessory for connecting the controller to a PC via the serial interface.

**Bootstrap protocol (BootP)**

In an Ethernet network, BootP is used to assign an IP address to a BootP client using a BootP server. For this example (default upon delivery for the ILC 150 ETH), the ILC 150 ETH (BootP client) sends a Boot_Request as a broadcast in the network. The MAC address of the transmitter is sent with the Boot_Request to provide unique identification. If the BootP server has been activated in PC Worx Express, PC Worx Express responds with a Boot_Reply. PC Worx Express uses this Boot_Reply to inform the ILC 150 ETH of its IP address and subnet mask. Please ensure that:

– The BootP server knows the MAC address sent by the BootP client
– A corresponding IP address and subnet mask have been assigned in PC Worx Express for the MAC address

Once the IP data has been transferred to the ILC 150 ETH successfully, PC Worx Express sends a corresponding acknowledgment message.

**PC/network adapter**

To determine whether your network permits the IP settings used in the example project (see Figure 3-3 on page 3-4), proceed as follows:

- In the Windows Control Panel, check the settings for your PC network adapter.
- If necessary, adjust these settings so that the ILC 150 ETH can be accessed in your network using the IP address used in the example project.

If your network does not permit the use of the IP address used in the example project, adjust the settings in the project information accordingly (see Figure 3-3 on page 3-4).

If any modifications are made to the project information that affect the IP settings for the controller, a warning is displayed. However, the modification is not implemented automatically. When a new project is created, the default settings are specified under "IP Settings" (see Figure 3-3 on page 3-4).

## Assigning IP settings

To set the IP address in PC Worx/PC Worx Express, proceed as described below.

Please note that by default upon delivery Inline BootP is active on the controller.

The IP address that is assigned here for the controller is also implemented as the IP address for the communication path via TCP/IP.

After assigning the IP address, PC Worx Express automatically creates a link via TCP/IP as a communication path to the Inline controller.

- Establish an Ethernet connection between your PC and the controller.
- In the PC Worx Express menu bar, select the "Extras, BootP/SNMP/TFTP-Configuration ..." menu.

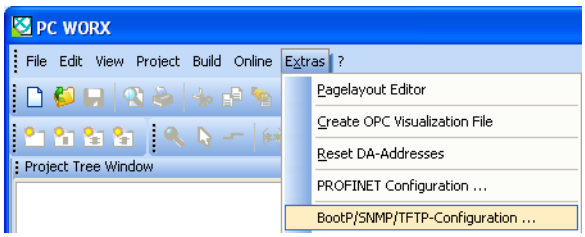



Figure 3-1 "Extras, BootP/SNMP/TFTP Configuration..." menu

- Activate the "BootP Server active" checkbox.

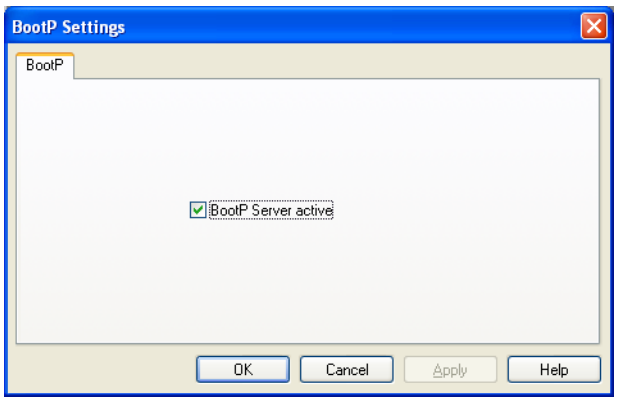



Figure 3-2 "BootP Server active" checkbox

- Switch to the bus configuration workspace.
- Select the controller node (here: "ILC 150 ETH").
- Select the "IP Settings" tab in the "Device Details" window.
- Enter the MAC address of the controller (see Figure 3-3 on page 3-4). It is printed on the device and starts with 00.A0.45.

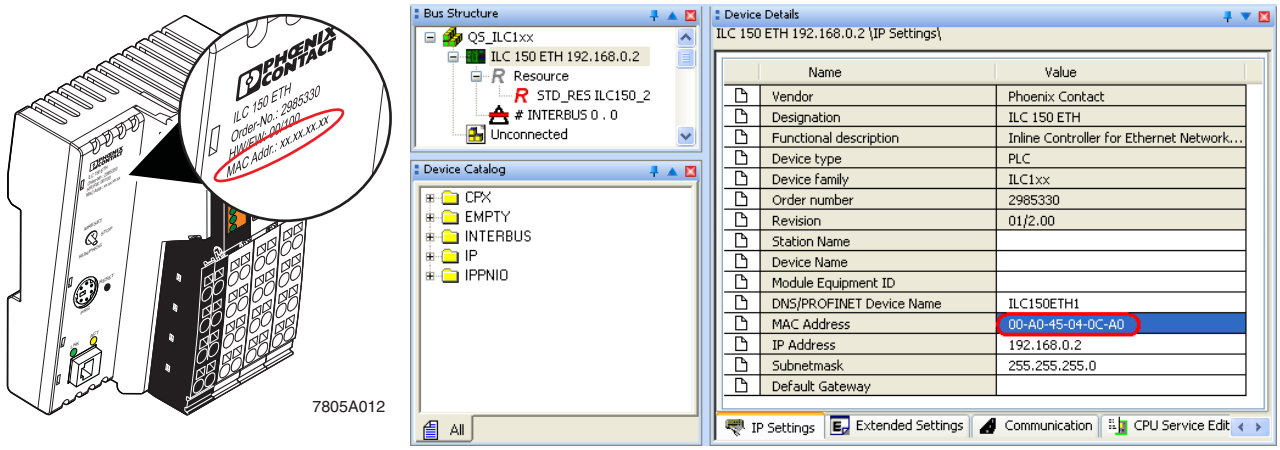



Figure 3-3 Entering the MAC address

- Perform a cold restart for the controller. To do this, switch the supply voltage off and then on again after about 2 seconds.

The controller is assigned the IP address, which is specified in the project for the controller. The following message appears in the message window in the "Bus Configurator" tab.

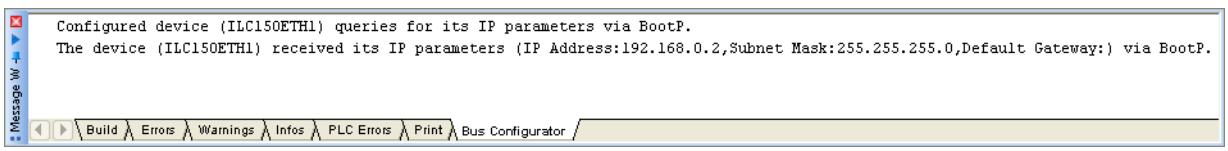



Figure 3-4 Message window following BootP

The IP address is now permanently stored on the controller Flash memory.

For additional information about setting the IP address with PC Worx/PC Worx Express, please refer to the quick start guides for the software used, the ILC 130 starter kit, and the ILC 150 construction kit.

## 3.3 ILC 170 ETH 2TX (ILC 190 ETH 2TX) as a PROFINET IO device

Please note that you can integrate the Inline controllers ILC 170 ETH 2TX and ILC 190 ETH 2TX in the **PC Worx** software as a **PROFINET IO device** only from the version listed in Section “Software version” on page 3-1 onwards.

In the PC Worx Express software you can switch on PROFINET IO device functions of the Inline controllers from the version listed in Section “Software version” on page 3-1 onwards.

This section uses an example to describe how to integrate the ILC 170 ETH 2TX as a PROFINET IO device. This procedure also applies to the ILC 190 ETH 2TX.

### Switching on the PROFINET IO IO device function of the ILC 170 ETH 2TX

You can switch on the PROFINET IO device function, after you have assigned an IP address for the ILC 170 ETH 2TX according to the description given in Section “Assigning the IP address for the controller/ BootP server” on page 3-2 (in the following example: 192.168.0.10).

- Select the "Status IO Device" item in the device details under "Network Settings".
- Under "Settings", select "activated" in the pull-down menu.

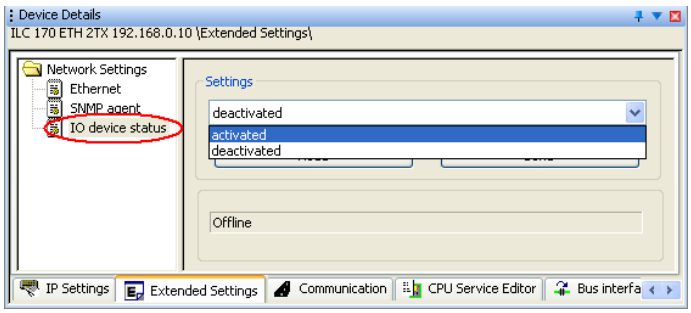



Figure 3-5 Network settings: Activating Status IO Device

- Click on "Send".
- In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

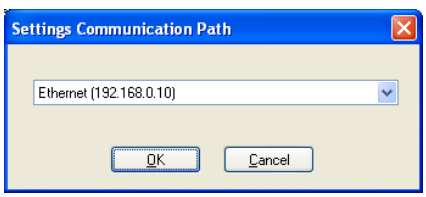



Figure 3-6 "Settings Communication Path" dialog box

Successful execution of the service will be displayed in the status window.

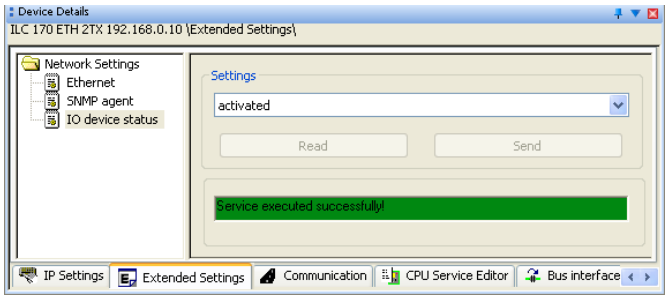



Figure 3-7 Status IO Device/Send settings: Service executed successfully

To transfer the network settings you have to reset the IO controller.

- Select the "Ethernet" item in the Device Details window under "Network Settings".
- In the "Activate Network Settings" area click the "Reset Control System" button.

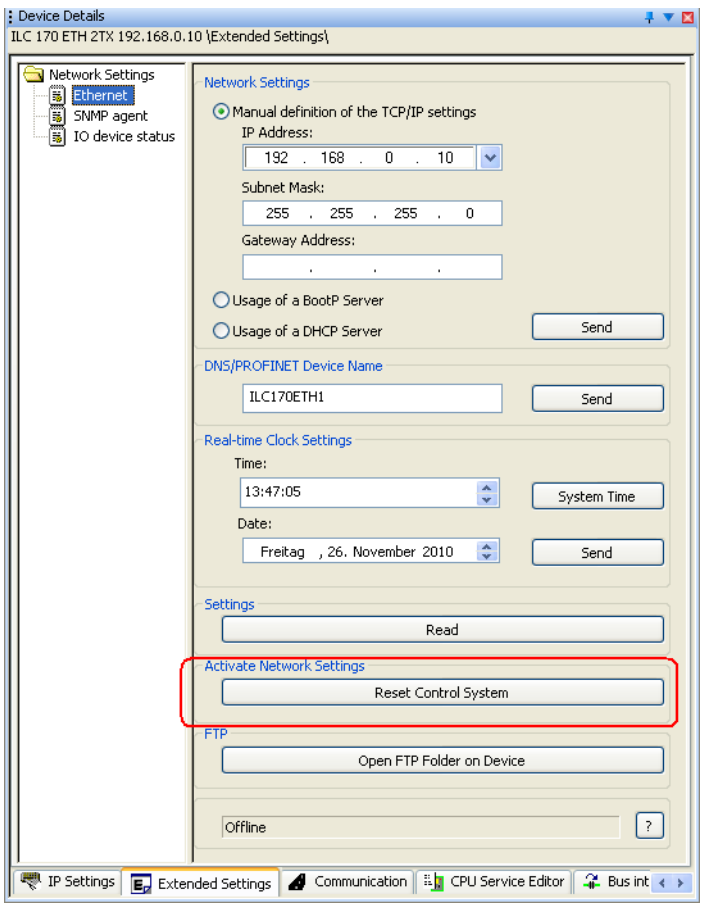



Figure 3-8 Activate Network Settings: Reset Control System

- In the "Settings Communication Path" dialog confirm with "OK" the suggested IP address or the one you have selected for your application.

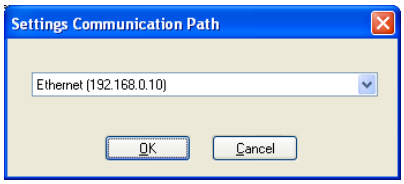



Figure 3-9 "Settings Communication Path" dialog box

Successful execution of the service will be displayed in the status window.

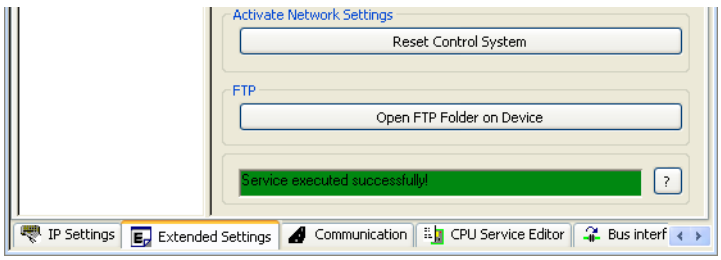



Figure 3-10 Activate Network Settings: Service executed successfully

Now you can read in the ILC 170 ETH 2TX as a PROFINET device in a PC Worx project.

## Integrating the ILC 170 ETH 2TX as a PROFINET IO device

The following section describes how you read the ILC 170 ETH 2TX as a PROFINET device in the PC Worx software.

As an alternative you can create the PC Worx project by selecting the devices from the device catalog. For more information on creating a PC Worx project, please refer to the online help or the software's quick start guide.

The following conditions apply for the example project:

- Higher-level controller; RFC 470 PN 3TX
- Settings of the controller:
  - IP address LAN1.1/1.2: 192.168.0.2
  - Subnet mask: 255.255.255.0
  - PROFINET device name: rfc-470-pn-1-ctrl
- Settings of the ILC 170 ETH 2TX as a PROFINET IO device:
  - IP address 192.168.0.10
  - Subnet mask: 255.255.255.0
  - PROFINET device name: ilc-170-pnd-18-8a-a8

You can read in the ILC 170 ETH 2TX as a PROFINET device after you have done the following:

- Switched on the PROFINET IO device function of the ILC 170 ETH 2TX
- Installed the PROFINET IO controller and the PROFINET devices (ILC 170 ETH 2TX and other IO devices according to your application)
- Created a project in PC Worx
- Configured the PROFINET IO controller according to your application
- Select "Read PROFINET ..." in the PROFINET context menu of the Bus Structure window.

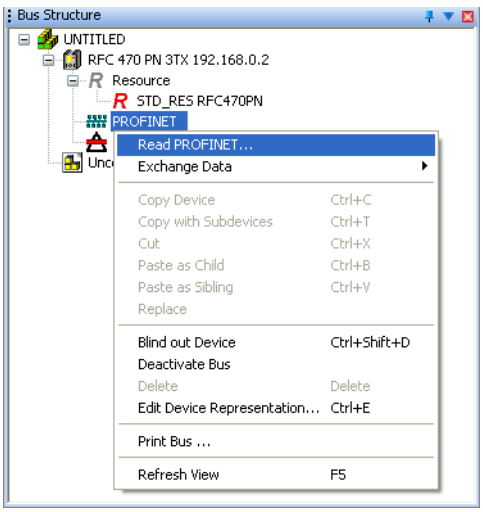



Figure 3-11 Bus Structure: PROFINET context menu "Read PROFINET"

The "Read PROFINET" dialog that opens up, shows the PROFINET devices that have been detected in the connected network.

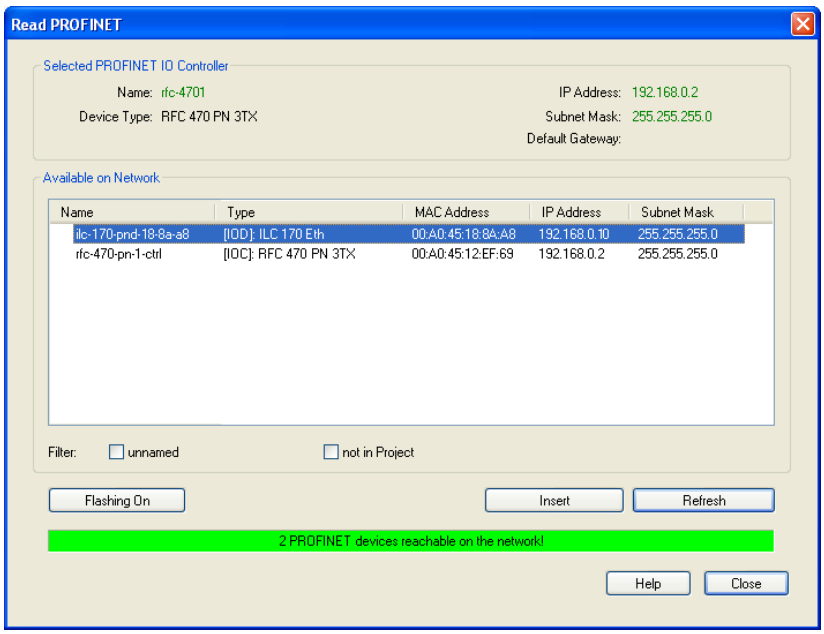



Figure 3-12 "Read PROFINET" dialog box

- Select the ILC 170 ETH 2TX and insert it as a PROFINET device by clicking on the "Insert" button.
- Close the dialog box by clicking on the "Close" button.

The PROFINET device inserted before will be displayed in the Bus Structure window.

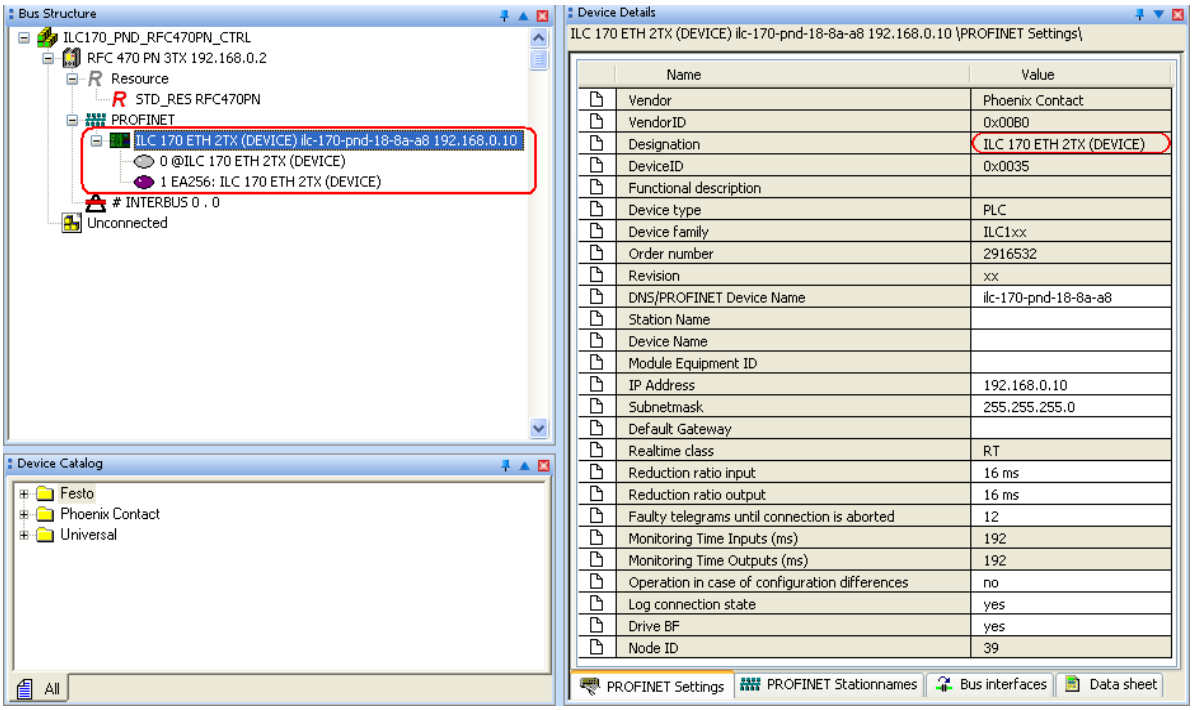



Figure 3-13 ILC 170 ETH 2TX inserted as a PROFINET IO device

The process data of the PROFINET device will be displayed in the Device Details window under the "Process Data" tab.

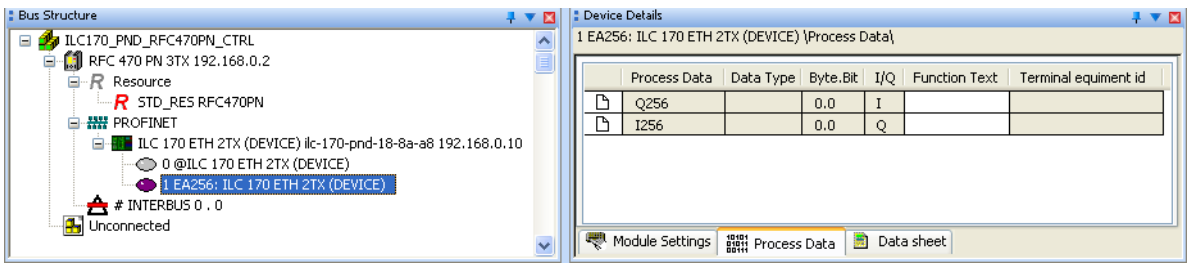



Figure 3-14 ILC 170 ETH 2TX as a PROFINET IO device: Process Data

The ILC 170 ETH 2TX is now available as a PROFINET IO device in the PC Worx project.

## 3.4 Setting the realtime clock under PC Worx Express

The procedure for setting the realtime clock is essentially the same in PC Worx and PC Worx Express. The following example describes the setting in PC Worx Express.

The time and date for the internal system clock of the Inline controller can be set under "Extended Settings" in the "Device Details" window with PC Worx Express.

To set the realtime clock, proceed as described in the quick start guide for the PC Worx Express version used.

## 3.5 Download changes

The "Download Changes" function supports the following Inline controllers from the specified device versions together with the specified versions of the PC Worx/PC Worx Express software tools.

Table 3-2 Version information for "Download Changes"

| Inline controller | Firmware version | PC Worx/PC Worx Express |
|---|---|---|
| ILC 130 ETH | ≥ 3.01 | ≥ 5.20 Service Pack 3 |
| ILC 150 ETH | ≥ 3.54 | ≥ 6.00 |
| ILC 155 ETH | ≥ 3.54 | ≥ 6.00 |
| ILC 170 ETH 2TX | ≥ 3.00 | ≥ 5.20 Service Pack 2 |
| ILC 190 ETH 2TX | ≥ 3.70 | ≥ 6.10 Service Pack 1 |

## 3.6 Parameterization memory and Internet Explorer

To delete files or store user-specific files on the internal parameterization memory, proceed as follows:

The FTP function must be activated in Internet Explorer. See also Section “Internet Explorer FTP function” on page 3-13.

- Switch to the bus configuration workspace in PC Worx.
- Select the control system, e.g.,"ILC 150 ETH", in the "Bus Structure" window.
- Select the "Extended Settings" tab in the "Device Details" window.
- Open Internet Explorer from this window by clicking on the "Open FTP folder on device" button.

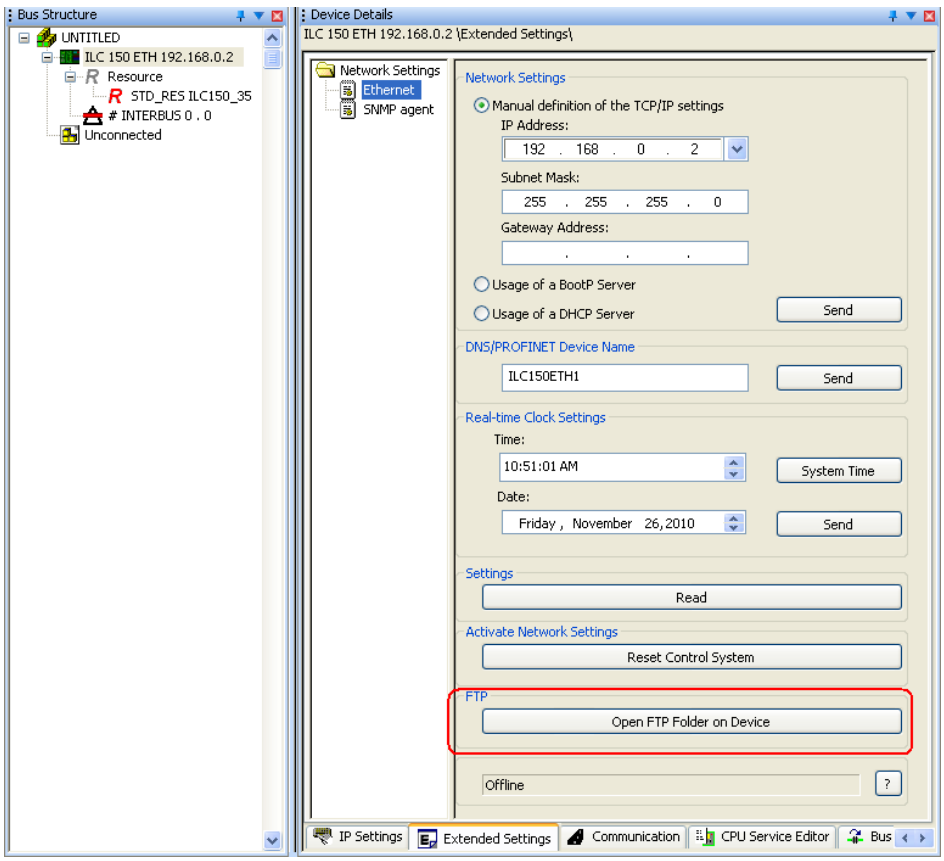



Figure 3-15 Extended Settings: Open FTP folder on device

The file structure, which is stored in the parameterization memory, is displayed in the Internet Explorer window.

Data may only be copied or deleted in the parameterization memory. Do not edit any files as Internet Explorer does not store modified data.

For the current state to be displayed, update the display after every action by means of the "View, Refresh" command.

### 3.6.1 Internet Explorer FTP function

- Activate this setting in Internet Explorer under "Tools, Internet Options..., Advanced".

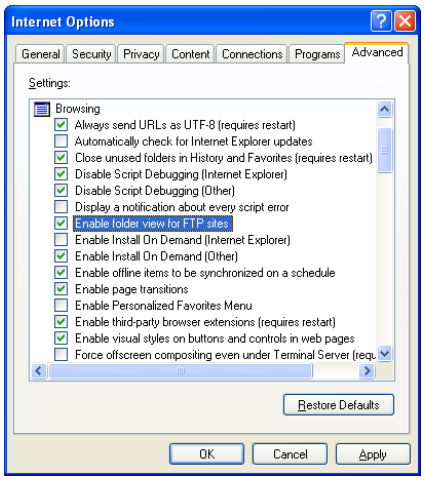



Figure 3-16 Internet Options: Enable folder view for FTP sites

### 3.6.2 Activating/deactivating the FTP server

To protect the Inline controller against unauthorized access, it may be necessary to deactivate the FTP server. The CPU_Set_Value_Request service with Var ID $0172_{hex}$ is used for this. This service activates or deactivates the FTP server for the runtime of the firmware. The set FTP server state is stored retentively and mapped to the ETH_SRV_FTP_ACTIVE system variables. This setting is restored the next time the Inline controller is rebooted.

Value range for the CPU_Set_Value_Request service:

| | | |
|---|---|---|
| Var ID | $0172_{hex}$ | |
| Value | $0000_{hex}$ | Deactivate FTP server |
| | $0001_{hex}$ | Activate FTP server |

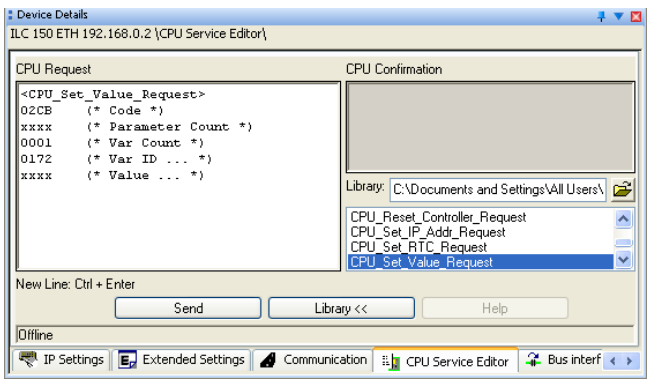



Figure 3-17 Deactivating the FTP server

### 3.6.3 Activating/deactivating the HTTP server

To protect the Inline controller against unauthorized access, it may be necessary to deactivate the HTTP server. The CPU_Set_Value_Request service with Var ID $0173_{hex}$ is used for this. This service activates or deactivates the HTTP server for the runtime of the firmware. The set HTTP server state is stored retentively and mapped to the ETH_SRV_HTTP_ACTIVE system variable. This setting is restored the next time the Inline controller is rebooted.

Value range for the CPU_Set_Value_Request service:

| | | |
|---|---|---|
| Var ID | $0173_{hex}$ | |
| Value | $0000_{hex}$ | Deactivate HTTP server |
| | $0001_{hex}$ | Activate HTTP server |

## 3.7 Activating/deactivating the journaling function

The file system of the parameterization memory supports the journaling function. As a result, voltage failures during write processes to the file system are tolerated by the Inline controller, however, the access speed becomes slower.

The journaling function is deactivated by default.

It can be activated or deactivated via the CPU_Set_Value_Request service with VarID 0194.

Value range for the CPU_Set_Value_Request service:

| | | |
|---|---|---|
| Code | 02CBhex | |
| Var Count | $0001_{hex}$ | |
| Var ID | $0194_{hex}$ | |
| Value | $0001_{hex}$ | Activate journaling function |
| | $0000_{hex}$ | Deactivate journaling function |

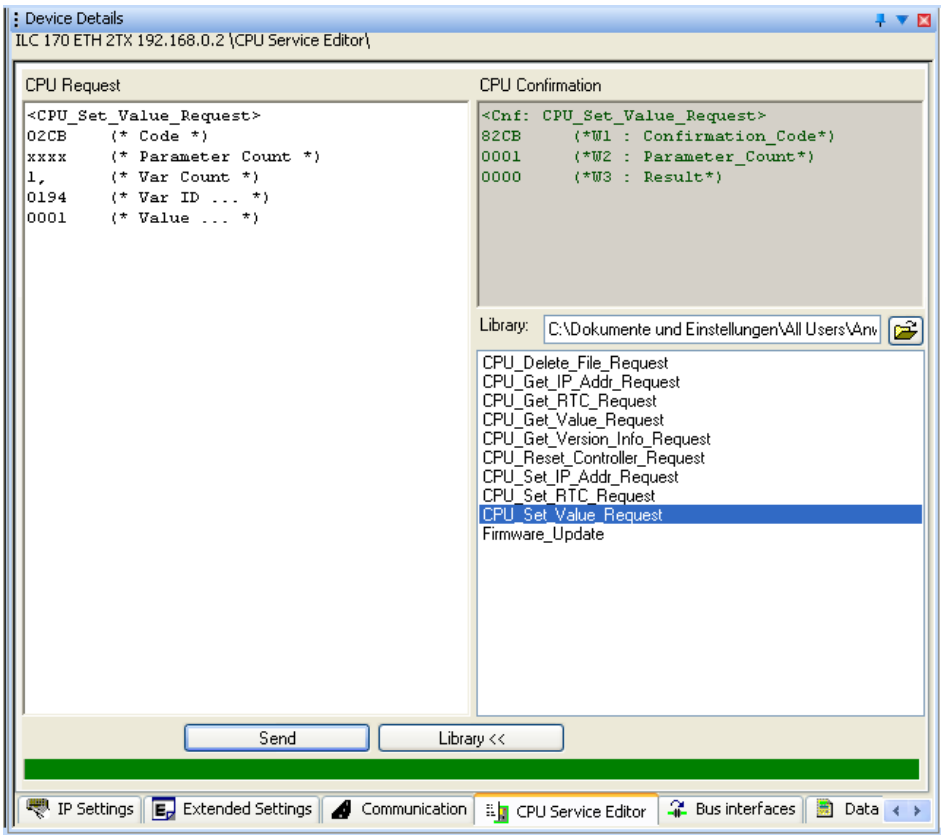



Figure 3-18 Activate journaling function

## 3.8 Function blocks for handling files in the parameterization memory

The function blocks are used to access files from within the application program. Some of the blocks support multiple instantiation. This means that it is possible to work with a number of different files within the same project. The blocks perform the standard functions that are required for typical file access operations.

The FILE_NOTIFY block is available in addition to the blocks for typical file access operations. This block can be used to detect files that have been modified in a directory containing user files. These modifications may include:

– Deletion of one or more files
– Writing one or more new files
– Modification of one or more existing files

Both modifications made to this directory via FTP (remote) and modifications made locally via function blocks or firmware services can be detected.

All file operations are subject to the following restrictions:
No directory hierarchies are supported. All file operations only affect the root directory of the parameterization memory.

The function blocks are valid for:

| Order designation | From hardware version | From firmware version |
|---|---|---|
| **ILC 130 ETH** | 01 | 3.01 |
| **ILC 150 ETH** | 02 | 2.10 |
| **ILC 155 ETH** | 01 | 2.04 |
| **ILC 170 ETH 2TX** | 01 | 3.00 |
| **ILC 190 ETH 2TX** | 00 | 3.70 |

Table 3-3 Overview of function blocks

| Function block | Short description |
|---|---|
| FILE_OPEN | Opens a file with a specific name |
| FILE_CLOSE | Closes a file with a specific handle |
| FILE_READ | Reads from a file with a specific handle |
| FILE_WRITE | Writes to a file with a specific handle |
| FILE_REMOVE | Deletes a file with a specific name |
| FILE_TELL | Determines the current position of the file pointer in a file |
| FILE_SEEK | Moves the current file pointer to a new position |
| FILE_NOTIFY | Displays files that have recently been created, deleted or modified |

The function blocks for handling files in the parameterization memory are described in the PC Worx online help.

## 3.9 Function blocks for Ethernet communication

The function blocks are used to establish Ethernet communication between two communication partners.

The IP communication blocks listed below enable IEC 61131-5-compliant communication between controllers via Ethernet or communication between controllers and Ethernet devices via TCP/IP or UDP/IP.

Implement all time and connection monitoring in the application program.

The function blocks are valid for the Inline controllers listed in the table below in the specified hardware/firmware versions and enable the specified maximum number of TCP/IP or UDP/IP communication connections:

| Order designation | Blocks | From hardware version (HW) | From firmware version (FW) | Ethernet connections to other communication partners (maximum) |
|---|---|---|---|---|
| **ILC 130 ETH** | IEC 61131-5 | 01 | 3.01 | 8 |
| | TCP/IP | | | |
| | UDP/IP | | | |
| **ILC 150 ETH** | IEC 61131-5 | 00 | 1.00 | 8 |
| | TCP/IP | | 1.00 | |
| | UDP/IP | | 2.00 | |
| **ILC 155 ETH** | IEC 61131-5 | 01 | 2.04 | 8 |
| | TCP/IP | | | |
| | UDP/IP | | | |
| **ILC 170 ETH 2TX** | IEC 61131-5 | 01 | 3.00 | 8 |
| | TCP/IP | 01 | 3.00 | 8 (HW 01/FW 3.00)<br>16 (HW 02/FW 3.54) |
| | UDP/IP | | | |
| **ILC 190 ETH 2TX** | IEC 61131-5 | 00 | 3.70 | 8 |
| | TCP/IP | | | 16 |
| | UDP/IP | | | |

Table 3-4 Overview of function blocks

| Function block | Short description |
|---|---|
| IP_CONNECT | Establishes a connection between two communication partners |
| IP_USEND | Transmits data to a communication partner |
| IP_URCV | Receives data from a communication partner |

The communication blocks are described in the PC Worx online help.
The extensions for the TCP/IP and UDP/IP function blocks are described in the "TCP/UDP COMMUNICATION" application note.

## 3.10 Function blocks for PCP communication

The function blocks are used to establish PCP communication between the Inline controller and PCP devices on INTERBUS.

The function blocks are valid for the Inline controllers listed in the table below in the specified HW/FW versions and enable the specified maximum number of PCP communication connections:

| Order designation | From hardware version (HW) | From firmware version (FW) | Connections to PCP devices (maximum) |
|---|---|---|---|
| **ILC 130 ETH** | 01 | 3.01 | 8 |
| **ILC 150 ETH** | 00<br>02 | 1.00<br>3.54 | 8<br>16 |
| **ILC 155 ETH** | 01<br>01 | 2.04<br>3.54 | 8<br>16 |
| **ILC 170 ETH 2TX** | 01 | 3.00 | 16 (HW 01/FW 3.00)<br>24 (HW 02/FW 3.54) |
| **ILC 190 ETH 2TX** | 00 | 3.70 | 24 |

Table 3-5 Overview of function blocks

| Function block | Short description |
|---|---|
| PCP_CONNECT | This block can be used to set up communication connections to each PCP device on INTERBUS. |
| PCP_WRITE | This block enables PCP objects to be written. |
| PCP_READ | This block enables data to be read from PCP objects. |
| PCP_SERVER | This block enables PCP service indications to be received and responses to be sent. |

The communication blocks are described in the PC Worx online help.

## 3.11 Alignment

The alignment of the data elements in the Inline controller memory can result in "data gaps" when storing data in the memory. The compiler automatically fills these gaps with padding bytes during the compiler process in order to prevent incorrect processing.

The disadvantage of the "automatic" filling of data gaps becomes apparent when data is transmitted from the Inline controller to another controller. If this controller does not know the memory algorithm of the Inline controller it will interpret the received data incorrectly.

It is therefore useful to program the filling of data gaps in your application program. Data transmissions to other controllers can thus be taken into consideration. For example, use byte arrays with an even number of bytes and/or word arrays in order to avoid data gaps in your application program.

The following should be taken into consideration when creating the program:

– Create data types in flat structures, i.e.,do not nest user-defined data types.
– Insert padding bytes manually in order to ensure the uniform size and layout of the data types.
– When inserting padding bytes, please observe the memory alignment method of the controllers used in the application (1-byte, 2-byte or 4-byte alignment).

**Program example with data gaps**

The following program example shows how data gaps are filled.

```
1   TYPE
2       Struct1 :
3       STRUCT
4           ByteElement :   BYTE;
5           WordElement :   WORD;
6       END_STRUCT:
7
8       Struct2 :
9       STRUCT
10          WordElement :   WORD;
11          ByteElement :   BYTE;
12      END_STRUCT:
13
14      Struct3 :
15      STRUCT
16          ByteElement1    :   BYTE;
17          ByteElement2    :   BYTE;
18      END_STRUCT:
19
20      Struct4 :
21      STRUCT
22          Struct2Element  :   Struct2;
23          Struct3Element  :   Struct3;
24      END_STRUCT:
25
26      Array1  :   ARRAY   [0..1]  OF  Struct2;
27  END_TYPE
```

Figure 3-19 Example programming

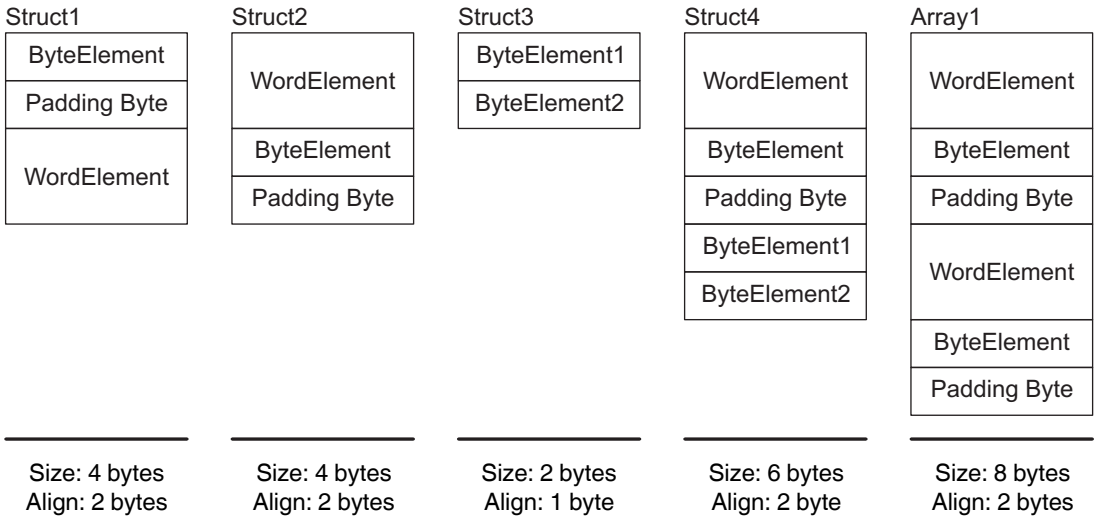



Figure 3-20 Alignment - padding bytes in data gaps

Struct1 receives a padding byte after the ByteElement so that the WordElement is at a WORD address (address that can be divided by 2 leaving no remainder). The alignment of the overall structure is based on the data type used with maximum alignment. In this case the WordElement specifies the alignment.

The size of Struct2 is calculated based on the elements used and the resulting alignment. The corresponding number of padding bytes is inserted so that the size of the data type with the value of the alignment can be divided by 2 leaving no remainder (data type size modulo alignment = 0).

Struct3 does not receive any padding bytes as the maximum alignment corresponds to one byte.

Due to the padding bytes that belong to the Struct2 structure, the Struct3 structure starts at an even address in Struct4.

Array1 receives 2 padding bytes, which corresponds to two consecutive Struct2 structures.

**Program example without data gaps**

The following program shows an example of how the filling of data gaps may appear in your application program. Fill data gaps, which are to be expected due to the memory alignment, with application data (padding bytes in Figure 3-21).

```
1   TYPE
2       Struct1 :
3       STRUCT
4           ByteElement :   BYTE;
5           ByteElement :   BYTE;   (*Padding-Byte*)
6           WordElement :   WORD;
7       END_STRUCT:
8
9       Struct2 :
10      STRUCT
11          WordElement :   WORD;
12          ByteElement :   BYTE;
13          ByteElement :   BYTE;   (*Padding-Byte*)
14      END_STRUCT:
15
16      Struct3 :
17      STRUCT
18          ByteElement1    :   BYTE;
19          ByteElement2    :   BYTE;
20      END_STRUCT:
21
22      STRUCT4 :
23      STRUCT
24          Struct2Element  :   Struct2;
25          Struct3Element  :   Struct3;
26      END_STRUCT:
27
28      Array1  :   ARRAY   [0..1]  OF  Struct2;
29  END_TYPE
```

Figure 3-21 Example programming with padding bytes

# 4 System variables and status information

## 4.1 General information

This section describes the special program functions of the PC Worx/PC Worx Express software that are available for the Inline controller.

The following descriptions of system variables and status information apply to PC Worx and PC Worx Express.

The Inline controller has a register set, which is used for diagnostics and easy control of the bus system. The diagnostic data is stored in the diagnostic status register and the diagnostic parameter register. These registers are available to the application program as system variables (system flags, global variables).

Operating states, error states, and additional information about the INTERBUS system can be evaluated in the application program.

For additional information on diagnostics, please refer to the following user manual:
– INTERBUS diagnostics guide
IBS SYS DIAG DSC UM E Order No. 2747293

## 4.2 Status register for local digital inputs and outputs

The following system variables can be used to read the local digital input and output states and to write the local digital output states.

Table 4-1 System variables of the status register for local digital inputs and outputs

| System variable | Type | Meaning |
|---|---|---|
| ONBOARD_INPUT | WORD | State of all local inputs |
| ONBOARD_INPUT_BIT0 | BOOL | State of local input IN1 |
| ONBOARD_INPUT_BIT1 | BOOL | State of local input IN2 |
| ONBOARD_INPUT_BIT2 | BOOL | State of local input IN3 |
| ONBOARD_INPUT_BIT3 | BOOL | State of local input IN4 |
| ONBOARD_INPUT_BIT4 | BOOL | State of local input IN5 |
| ONBOARD_INPUT_BIT5 | BOOL | State of local input IN6 |
| ONBOARD_INPUT_BIT6 | BOOL | State of local input IN7 |
| ONBOARD_INPUT_BIT7 | BOOL | State of local input IN8 |
| ONBOARD_OUTPUT_BIT0 | BOOL | State of local output OUT1 |
| ONBOARD_OUTPUT_BIT1 | BOOL | State of local output OUT 2 |
| ONBOARD_OUTPUT_BIT2 | BOOL | State of local output OUT3 |
| ONBOARD_OUTPUT_BIT3 | BOOL | State of local output OUT4 |
| ONBOARD_OUTPUT_OVERLOAD_0_3 | BOOL | One local output overloaded |

## 4.3 Diagnostic status register

In the diagnostic status register, information on the operating state of the Inline controller are stored. Every bit in the diagnostic status register is assigned a certain Inline controller state.

The following system variables can be used to read the diagnostic status register information.

Table 4-2 System variables of the diagnostic status register

| System variable | Type | Meaning |
|---|---|---|
| MASTER_DIAG_STATUS_REG_USER | BOOL | User error/parameterization error |
| MASTER_DIAG_STATUS_REG_PF | BOOL | Peripheral fault |
| MASTER_DIAG_STATUS_REG_BUS | BOOL | Bus error |
| MASTER_DIAG_STATUS_REG_CTRL | BOOL | Error on the Inline controller/hardware fault |
| MASTER_DIAG_STATUS_REG_DTC | BOOL | Diagnostic routine active |
| MASTER_DIAG_STATUS_REG_RUN | BOOL | Data transmission active |
| MASTER_DIAG_STATUS_REG_ACT | BOOL | Selected configuration is ready for operation |
| MASTER_DIAG_STATUS_REG_RDY | BOOL | Inline controller ready to operate |
| MASTER_DIAG_STATUS_REG_BSA | BOOL | Bus segment aborted |
| MASTER_DIAG_STATUS_REG_SYSFAIL | BOOL | System failure |
| MASTER_DIAG_STATUS_REG_RES | BOOL | Standard function processed negatively |
| MASTER_DIAG_STATUS_REG_SYNCRES | BOOL | Synchronization error occurred |
| MASTER_DIAG_STATUS_REG_DCR | BOOL | Faulty data cycles |
| MASTER_DIAG_STATUS_REG_WARN | BOOL | Defined warning time exceeded |
| MASTER_DIAG_STATUS_REG_QUAL | BOOL | Defined error density exceeded |
| MASTER_DIAG_STATUS_REG_SSINFO | BOOL | Pending message |
| MASTER_DIAG_STATUS_REG_HI | BYTE | Master diagnostic status register, high byte |
| MASTER_DIAG_STATUS_REG_LOW | BYTE | Master diagnostic status register, low byte |

## 4.4 Diagnostic parameter register

In the diagnostic parameter register, you will find additional information on the error indicated in the diagnostic status register. The following information is stored in the diagnostic parameter register:

– Error location
– Error code

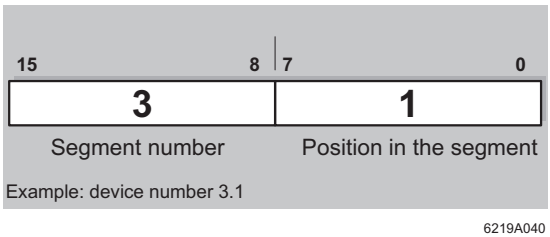



Figure 4-1 Error location in the diagnostic parameter register

**Special case:** If an interface error cannot be located, the value 128 is indicated in the diagnostic parameter register, i.e., bit 7 is set.

The diagnostic parameter register is rewritten whenever an error occurs. The diagnostic parameter register contains the value "0" if no errors are detected.

Table 4-3 System variables of the diagnostic parameter register

| System variable | Type | Meaning |
|---|---|---|
| MASTER_DIAG_PARAM_REG_HI | BYTE | Diagnostic parameter register, high byte |
| MASTER_DIAG_PARAM_REG_LOW | BYTE | Diagnostic parameter register, low byte |
| MASTER_DIAG_PARAM_2_REG_HI | BYTE | Extended diagnostic parameter register, high byte |
| MASTER_DIAG_PARAM_2_REG_LOW | BYTE | Extended diagnostic parameter register, low byte |

## 4.5 PROFINET register (ILC 170 ETH 2TX and ILC 190 ETH 2TX)

Please observe that the following PROFINET registers are only available when you use Inline controllers with the specified device version and the following software versions or later:

| | |
|---|---|
| ILC 170 ETH 2TX | Hardware "02"/Firmware "3.50"<br>PC Worx Version 6.00 Service Pack 2 or later (part of the AUTOMATIONWORX Software Suite 2009 1.50 Service Pack 2) |
| ILC 190 ETH 2TX | Hardware "00"/Firmware "3.70"<br>PC Worx Version 6.10 Service Pack 1 or later (part of the AUTOMATIONWORX Software Suite 2010 1.60 Service Pack 1) |

The PC Worx Express software does not support this function.

Table 4-4 PROFINET system variables (PROFINET IO device function)

| System variable | Type | Meaning |
|---|---|---|
| PND_S1_PLC_RUN | BOOL | Status of the higher-level control system/ IO controllers |
| PND_S1_VALID_DATA_CYCLE | BOOL | The higher-level control system/ IO controller has established the connection. |
| PND_S1_OUTPUT_STATUS_GOOD | BOOL | IOP status of the higher-level control system/ IO controller |
| PND_S1_DATA_LENGTH | WORD | Process data length that was configured for the IO device. |
| PND_S1_OUTPUTS | PND_IO_256 | OUT process data<br>Memory area for OUT process data that the IO device sends to the higher-level control system/ IO controller. |
| PND_S1_INPUTS | PND_IO_256 | IN process data<br>Memory area for IN process data that the IO device receives from the higher-level control system/ IO controller. |
| PND_IO_DRIVEN_BY_PLC | INT | Applicative system redundancy:<br>Information from which higher-level IO controller the data in the IO device comes from (cf. Figure 2-5 on page 2-6).<br>0: No IO controller<br>1: Controller A<br>2: Controller B |

## 4.6 IEC -61131 runtime system

There is a separate group of variables for the IEC 61131 runtime system.

Table 4-5 System variables of the IEC 61131 runtime system

| System variable | Type | Meaning |
|---|---|---|
| PLCMODE_ON | BOOL | PLC status ON: The runtime system on the Inline controller is ready to operate. |
| PLCMODE_RUN | BOOL | PLC status OPERATION: The application program is running. |
| PLCMODE_STOP | BOOL | PLC status STOP: The application program is currently not running. |
| PLCMODE_HALT | BOOL | PLC status HALT: The application program was stopped at an unspecified point. |
| PLCDEBUG_BPSET | BOOL | Breakpoint set: At least one breakpoint has been set in the application program. |
| PLCDEBUG_FORCE | BOOL | Variable(s) forced: At least one variable is being continuously overwritten (forced). |
| PLCDEBUG_POWERFLOW | BOOL | Powerflow ON: In "Powerflow" mode, you can see which parts of your application program are being processed. This bit indicates whether "Powerflow" mode is active. |
| PLC_TICKS_PER_SEC | INT | System ticks per second: This variable shows how many pulses the system clock of the Inline controller delivers per second. |
| PLC_SYS_TICK_CNT | DINT | Number of system ticks: This variable shows the total number of pulses delivered by the system clock since the last startup. |
| PLC_TASK_AVAILABLE | INT | Number of available PLC tasks |
| PLC_SYSTASK_AVAILABLE | INT | Number of available system tasks |
| PLC_MAX_ERRORS | DINT | Maximum number of "errors, warnings and logging events" If this maximum number is reached, the controller is stopped. |
| PLC_ERRORS | DINT | Number of "errors, warnings and logging events" currently entered. |
| PLC_TASK_DEFINED | INT | Number of tasks used |
| PLC_TASK_1 | Record, elements = 17 | Information on task 1 |
| : | : | : |
| PLC_TASK_8 | Record, elements = 9 | Information on task 8 |

## 4.7 Control processor

The system variables listed below show the states of the diagnostic status register on the control processor of the Inline controller.

Table 4-6 System variables of the control processor

| System variable | Type | Meaning |
|---|---|---|
| COP_DIAG_STATUS_REG_RT_ERR | BOOL | A runtime error (out of realtime) has occurred on the control processor. |
| COP_DIAG_STATUS_REG_FAT_ERR | BOOL | A fatal error has occurred on the control processor. Division by zero, for example, leads to a fatal error. |
| COP_DIAG_STATUS_REG_WARN | BOOL | A warning has been issued on the control processor. |
| COP_DIAG_STATUS_REG_PON | BOOL | Power ON (COP): The control processor is ready to operate. |
| COP_DIAG_STATUS_REG_FC_RUN | BOOL | Runtime system RUN |
| COP_DIAG_STATUS_REG_FC_STOP | BOOL | Runtime system STOP |
| COP_DIAG_STATUS_REG_FC_HALT | BOOL | Runtime system HALT |
| COP_DIAG_STATUS_REG_FC_LDG | BOOL | Runtime system LOADING |
| COP_DIAG_STATUS_REG_FC_DBG | BOOL | Runtime system DEBUG |
| COP_DIAG_STATUS_REG_FC_RDO | BOOL | Runtime system READONLY |
| COP_DIAG_PARAM_REG | WORD | Diagnostic parameter register of the control processor |
| COP_DIAG_PARAM_2_REG | WORD | Extended diagnostic parameter register of the control processor |

The system variable given below shows status information of the control processor of the Inline controller.

Table 4-7 System variable of the control processor

| System variable | Type | Meaning |
|---|---|---|
| COP_CPU_LOAD_WARNING | BOOL | The control processor is reaching the limits of its capacity. |

## 4.8 Battery, realtime clock

Table 4-8 System variables of the battery and realtime clock

| System variable | Type | Meaning |
|---|---|---|
| RTC_BATTERY_LOW | BOOL | The realtime clock battery is low. |
| RTC_DATA_INVALID | BOOL | Realtime clock data is invalid. |

## 4.9 Power supplies

Table 4-9 System variables of the power supplies

| System variable | Type | Meaning |
|---|---|---|
| POWER_SUPPLY_MAIN_OK | BOOL | The 24 V main power supply is OK. |
| POWER_SUPPLY_INPUTS_OK | BOOL | The 24 V power supply for the local inputs is OK. |
| POWER_SUPPLY_OUTPUTS_0_3_OK | BOOL | The 24 V power supply for the local outputs is OK. (Bits 0 to 3) |

## 4.10 Mode selector switch

Table 4-10 System variables of the mode selector switch

| System variable | Type | Meaning |
|---|---|---|
| KEY_SWITCH_RESET | BOOL | The mode selector switch is in the MRESET position. |
| KEY_SWITCH_STOP | BOOL | The mode selector switch is in the STOP position. |
| KEY_SWITCH_RUN_PROG | BOOL | The mode selector switch is in the RUN_PROG position. |

## 4.11 System time

Table 4-11 System variables of the system time

| System variable | Type | Meaning |
|---|---|---|
| RTC_HOURS | INT | System time (hours) |
| RTC_MINUTES | INT | System time (minutes) |
| RTC_SECONDS | INT | System time (seconds) |
| RTC_DAY | INT | System time (day) |
| RTC_MONTH | INT | System time (month) |
| RTC_YEAR | INT | System time (year) |

# 5 Technical data and ordering data

## 5.1 Technical data

| General data | |
|---|---|
| Dimensions | 80 mm x 119.8 mm x 71.5 mm |
| Weight | 285 g, approximately(ILC 130 ETH, ILC 150 ETH, ILC 155 ETH)<br>295 g, approximately (ILC 170 ETH 2TX, ILC 190 ETH 2TX) |
| Connection data for connectors | |
| Connection method | Spring-cage connection |
| Conductor cross-section | |
| Single-wire/terminal point, solid | 0.08 $mm^2$ to 1.5 $mm^2$ |
| Single-wire/terminal point, stranded | 0.08 $mm^2$ to 1.5 $mm^2$ |
| Single-wire/terminal point, AWG | 28 to 16 |
| | We recommend using a conductor cross-section of 0.2 $mm^2$ to 1.5 $mm^2$. |

| General supply |
|---|

Use a power supply without fall-back characteristic curve (see Section “Sizing of the power supply” on page 2-26).

| 24 V main supply ($U_M$) | |
|---|---|
| Connection method | Spring-cage terminals |
| Nominal value | 24 V DC |
| Tolerance | -15%/+20% (according to EN 61131-2) |
| Ripple | ±5% |
| Current consumption at nominal voltage (typical) | 6 mA + 7 mA for each input set |
| Current consumption at nominal voltage (maximum) | 8 A DC |
| Continuation | Via potential routing |
| Protective measures | |
| Surge voltage | Input protective diodes (can be destroyed by permanent overload)<br>Pulse loads up to 1500 W are short circuited by the input protective diode. |
| Polarity reversal | Parallel diodes for protection against polarity reversal; in the event of an error the high current flowing through the diodes causes the preconnected fuse to blow. |

**NOTE:**

Provide external fuses for this 24 V area. The power supply unit must be able to supply 4 times (400%) the nominal current of the external fuse, to ensure that the fuse blows safely in the event of an error.

| 24 V segment supply $U_S$ | |
|---|---|
| Connection method | Spring-cage terminals |
| Nominal value | 24 V DC |
| Tolerance | -15%/+20% (according to EN 61131-2) |
| Ripple | ±5% |
| Current consumption at nominal voltage (typical) | 10 mA + 10 mA for each output set + load |
| Current consumption at nominal voltage (maximum) | 8 A |
| Continuation | Through potential routing |
| Protective measures | |
| Surge voltage | Input protective diodes (can be destroyed by permanent overload)<br>Pulse loads up to 1500 W are short circuited by the input protective diode. |
| Polarity reversal | Parallel diodes for protection against polarity reversal; in the event of an error the high current flowing through the diodes causes the preconnected fuse to blow. |

**NOTE:**

Provide external fuses for this 24 V area. The power supply unit must be able to supply 4 times (400%) the nominal current of the external fuse, to ensure that the fuse blows safely in the event of an error.

| 24 V supply $U_{ILC}$ | |
|---|---|
| Connection method | Spring-cage terminals |
| Nominal value | 24 V DC |
| Tolerance | -15%/+20% (according to EN 61131-2) |
| Ripple | ±5% |
| Permissible range | 19.2 V DC to 30 V DC |
| Current consumption at nominal voltage (minimum) | 80 mA (no-load operation, i.e., no local bus devices connected, bus inactive) |
| Current consumption at nominal voltage (typical) | 210 mA |
| Current consumption at nominal voltage (maximum) | 870 mA (370 mA communications power + 500 mA analog power supply) |
| Protective measures | |
| Surge voltage | Input protective diodes (can be destroyed by permanent overload) Pulse loads up to 1500 W are short circuited by the input protective diode. |
| Polarity reversal | Serial diode in the lead path of the power supply unit; in the event of an error only a low current flows. In the event of an error, no fuse trips within the external power supply unit |

**NOTE:**

**Observe the current consumption of the Inline terminals**

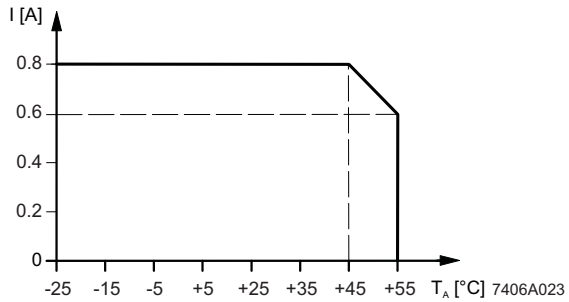
Observe the logic current consumption of each device when configuring an Inline station. It is specified in every terminal-specific data sheet. The current consumption can differ depending on the individual terminal. The permissible number of devices that can be connected therefore depends on the specific station structure.

**Protection of the external power supply unit**

Ensure protection of 2 A by fuses through the external power supply unit.

## 7.5 V communications power $U_L$ (potential jumper)

| | |
|---|---|
| Nominal value | 7.5 V DC |
| Tolerance | ±5% |
| Ripple | ±1.5% |
| Maximum output current | 0.8 A DC, observe derating (internally protected against short circuit) |
| Derating | |




I [A] Logic current in A
$T_A$ [°C] Ambient temperature in °C

## 24 V analog supply $U_{ANA}$ (potential jumper)

| | |
|---|---|
| Nominal value | 24 V DC |
| Tolerance | -15% / +20% |
| Ripple | ±5% |
| Maximum output current | 0.5 A DC, observe derating |
| Protective measures | Electric short-circuit protection |

## PROFINET (ILC 170 ETH 2TX and ILC 190 ETH 2TX)

| | |
|---|---|
| Type | PROFINET IO device |
| Specification | 2.2 |
| Performance class | RT |
| Update rate | ≥ 1 ms |
| Number of slots | 1 |
| Vendor ID | |
| ILC 170 ETH 2TX | $00B0_{hex}$ / $176_{dec}$ |
| ILC 190 ETH 2TX | $00B0_{hex}$ / $176_{dec}$ |
| Device ID | |
| ILC 170 ETH 2TX | $0035_{hex}$ / $53_{dec}$ |
| ILC 190 ETH 2TX | $0076_{hex}$ / $118_{dec}$ |

## INTERBUS

| | |
|---|---|
| Number of I/O points | 4096, maximum |
| Number of data words | 256, maximum |
| Transmission speed | 500 kbps or 2 Mbps |

This speed is automatically set according to the connected Inline terminals. Only use terminals with a uniform transmission speed in the entire connected Inline system (local bus and remote bus).

| | |
|---|---|
| Transmission reliability | CR check (hamming distance: 4) |
| Protocol | EN 50254 |

## Number of devices in the INTERBUS system

| | |
|---|---|
| Total number of bus devices | |
| ILC 130 ETH | 63, maximum |
| ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX | 128, maximum |
| Number of remote bus devices | |
| ILC 130 ETH | – |
| ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX | 32, maximum |
| Number of devices with parameter channel (PCP) | See Section 3.10 on page 3-18 |
| ILC 130 ETH, | 8 |
| ILC 150 ETH, | 8 (HW/FW: ≥ 00/1.00), 16 (HW/FW: ≥ 02/3.54) |
| ILC 155 ETH, | 8 (HW/FW: ≥ 01/2.04), 16 (HW/FW: ≥ 01/3.54) |
| ILC 170 ETH 2TX | 16 (HW/FW: ≥ 01/3.00), 24 (HW/FW: ≥ 02/3.54) |
| ILC 190 ETH 2TX | 24 |
| Number of remote bus levels | See Section 2.2 on page 2-3 |
| ILC 130 ETH | – |
| ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX | 4, maximum |

Please note that the ILC 130 ETH does not support connection of the INTERBUS remote bus.

## Network interface

| | |
|---|---|
| Type | |
| ILC 130 ETH, ILC 150 ETH, ILC 155 ETH | 1 x Ethernet; 10 BASE-T and 100 BASE-T(X) |
| ILC 170 ETH 2TX, ILC 190 ETH 2TX | 2 x Ethernet; 10 BASE-T and 100 BASE-T(X) |
| Transmission speed | 10 Mbps (10 BASE-T), 100 Mbps (100 BASE-T(X))<br>half duplex, full duplex, auto negotiation |

This speed cannot be set manually. It is set automatically by means of auto negotiation.

| | |
|---|---|
| Connection method | CAT5 twisted-pair cable<br>Twisted-pair cable with a conductor cross-section of 0.14 $mm^2$ to 0.22 $mm^2$<br>8-pos. RJ45 female connector |

## Inline local bus

| | |
|---|---|
| Interface | Inline local bus |
| Electrical isolation | No |
| Number of devices | |
| ILC 130 ETH | 63, maximum |
| ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX | 128, maximum |

**NOTE:** Observe the current consumption of the Inline terminals

Observe the logic current consumption of each device when configuring an Inline station. It is specified in every terminal-specific data sheet. The current consumption can differ depending on the individual terminal. The permissible number of devices that can be connected therefore depends on the specific station structure.

## Diagnostic interface

| | |
|---|---|
| Connection method | 6-pos. mini-DIN female connector (PS/2) |
| Interface type | RS-232 |
| Transmission rate | 9600 baud |
| Electrical isolation | No |

## Digital inputs

| | |
|---|---|
| Number | 8 |
| Input design | According to EN 61131-2 Type 1 |
| Definition of the switching threshold | |
| Maximum low-level voltage | 5 V DC |
| Minimum high-level voltage | 15 V DC |
| Nominal input voltage | 24 V DC |
| Permissible range | -0.5 V < $U_{IN}$ < +30 V DC |
| Nominal input current at 24 V | 7 mA, typical; 15mA, maximum |
| Hardware filter times (typical) | |
| Inputs I1 to I8 | |
| Signal change 0 -> 1<br>Signal change 1 -> 0 | 5 ms<br>5 ms |
| Permissible cable length to the sensor | 30 m (to ensure conformance with EMC Directive 89/336/EEC) |
| Use of AC sensors | AC sensors in the voltage range < $U_{IN}$ are limited in application (according to the input design) |

## Digital outputs

| | |
|---|---|
| Number | 4 |
| Output design | Protected outputs according to EN 61131-2 |
| Nominal output voltage | 24 V DC |
| Nominal output current | 500 mA |

## Diagnostic and status indicators

| | |
|---|---|
| IEC 61131 runtime system (PLC) | FR, FF |
| Ethernet (ETH) | LINK, ACT |
| INTERBUS diagnostics (IL) | RDY, BSA, FAIL, PF |

| Diagnostic and status indicators | |
|---|---|
| Digital inputs and outputs | I1 to I8, E, Q1 to Q4 |
| Supply voltages | US, UM, UL |
| PROFINET (ILC 170 ETH 2TX, ILC 190 ETH 2TX) | BF, SF |

| IEC 61131 runtime system | ILC 130 ETH | ILC 150 ETH | ILC 155 ETH, ILC 170 ETH 2TX | ILC 190 ETH 2TX |
|---|---|---|---|---|
| Programming system | PC Worx | PC Worx | PC Worx | PC Worx |
| Processing speed | | | | |
| 1 K mix instructions | 1.7 ms | 1.5 ms | 1.5 ms | 1.3 ms |
| 1 K bit instructions | 90 µs | 90 µs | 90 µs | 90 µs |
| Shortest cycle time (for cyclic task) | 1 ms | 1 ms | 1 ms | 1 ms |
| Program memory | 192 KB, 16 K instructions (IL) | 256 KB, 21 K instructions (IL) | 512 KB, 43 K instructions (IL) | 1 MB, 86 K instructions (IL) |
| Data memory | 192 KB | 256 KB | 512 KB | 1 MB |
| Retentive data memory | 8 KB, NVRAM | 8 KB, NVRAM | 48 KB, NVRAM | 48 KB, NVRAM |
| Number of data blocks | Depends on data memory | Depends on data memory | Depends on data memory | Depends on data memory |
| Number of control tasks | 8 | 8 | 8 | 8 |
| Parameterization memory | | | | |
| Integrated (ILC 130 ETH, ILC 150 ETH, ILC 155 ETH) | 4 MB Flash memory (100,000 write access operations per sector, typical) | 4 MB Flash memory (100,000 write access operations per sector, typical) | 4 MB Flash memory (100,000 write access operations per sector, typical) | – |
| Plug-in, SD card (ILC 170 ETH 2TX, ILC 190 ETH 2TX) | – | – | 256 MB (1,000,000 write access operations per sector, typical), see Section “Accessories” on page 5-8 | 256 MB (1,000,000 write access operations per sector, typical), see Section “Accessories” on page 5-8 |

Please note that the number of write access operations to the parameterization memory is limited.

We recommend that you limit the number of write access operations on the parameterization memory in your application program by first storing data on the data memory and/or the memory for retentive data (NVRAM). Write access operations on the parameterization memory for small volumes of data (bit, byte) should not be possible in your application program. Data should only be transferred from the data memory/memory for retentive data to the parameterization memory if the data memory/memory for retentive data is full, i.e., data must be deleted first to enable further write access operations.

| Real-time clock | |
|---|---|
| Accuracy | 1 minute/week, maximum |
| Power reserve | 3 days |

| Ambient conditions | |
|---|---|
| Degree of protection | IP20 (EN 60529:1991) |
| Permissible temperature (operation) | -25°C to +55°C |
| Permissible temperature (storage/transport) | -25°C to +85°C |

This temperature range is only guaranteed if the Inline controller is mounted horizontally.

| | |
|---|---|
| Permissible humidity (operation/storage/transport) | 10% to 95% according to DIN EN 61131-2 |
| Permissible air pressure (operation/storage/transport) | 70 kPa to 106 kPa (up to 3000 m above sea level) |

## Mechanical tests

| | |
|---|---|
| Vibration resistance according to EN 60068-2-6, IEC 60068-2-6 | Operation: 5g |
| Shock test according to EN 60068-2-27, IEC 60068-2-27 | 25g |

## Conformance with EMC directive 2004/108/EC

### Noise immunity test according to EN 61000-6-2

| | | |
|---|---|---|
| Electrostatic discharge (ESD) | EN 61000-4-2/<br>IEC 61000-4-2 | Criterion B<br>6 kV contact discharge<br>8 kV air discharge |
| Electromagnetic fields | EN 61000-4-3<br>IEC 61000-4-3 | Criterion A,<br>Field strength 10 V/m |
| Fast transients (burst) | EN 61000-4-4/<br>IEC 61000-4-4 | Criterion B<br>Supply lines: 2 kV<br>Signal/data lines: 2 kV |
| Surge test | EN 61000-4-5<br>IEC 61000-4-5 | Criterion B<br>Signal/data lines: 1 kV<br>Supply lines: 0.5 kV |
| Conducted interference | EN 61000-4-6<br>IEC 61000-4-6 | Criterion A<br>Test voltage 10 V |

### Noise emission test according to EN 61000-6-4

| | | |
|---|---|---|
| Noise emission of housing | EN 55011 | Class A |

## Approvals

For the latest approvals, please visit www.phoenixcontact.com or www.phoenixcontact.net/catalog.

# 5.2 Ordering data

## 5.2.1 Modules

| Description | Type | Order No. | Pcs./Pkt. |
|---|---|---|---|
| Inline controller, including accessories (connectors and labeling fields) | ILC 130 ETH | 2988803 | 1 |
| Inline controller, including accessories (connectors and labeling fields) | ILC 150 ETH | 2985330 | 1 |
| Inline controller, including accessories (connectors and labeling fields) | ILC 155 ETH | 2988188 | 1 |
| Inline controller, including accessories (connectors and labeling fields) | ILC 170 ETH 2TX | 2916532 | 1 |
| Inline controller, including accessories (connectors and labeling fields) | ILC 190 ETH 2TX | 2700527 | 1 |

## 5.2.2 Accessories

| Description | Type | Order No. | Pcs./Pkt. |
|---|---|---|---|
| Inline connector set for Inline bus coupler with connected I/Os | IL BKDIO-PLSET | 2878599 | 1 |
| Connecting cable for connecting the Inline controller to a PC (V.24 (RS232) cable) | PRG CAB MINI DIN | 2730611 | 1 |
| 256 MB SD memory card | SD FLASH 256MB | 2988120 | 1 |
| Program and configuration memory, plug-in, 256 Mbytes with license key for function block libraries, e.g., for: SNMP, SQL, wireless, motion functions, etc | SD FLASH 256MB APPLIC A | 2988816 | 1 |
| Program and configuration memory, plug-in, 2 GByte | SD FLASH 2GB | 2988162 | 1 |
| Program and configuration memory, plug-in, 2 Gbytes with license key for function block libraries, e.g., for: SNMP, SQL, wireless, motion functions, etc. | SD FLASH 2GB APPLIC A | 2701190 | 1 |
| QUINT POWER power supply units | See latest Phoenix Contact INTERFACE catalog | | |

## 5.2.3 Software

| Description | Type | Order No. | Pcs./Pkt. |
|---|---|---|---|
| PC Worx Express automation software | PC WORX EXPRESS | 2988670 | 1 |
| PC Worx automation software | See latest Phoenix Contact AUTOMATION catalog | | |
| Configuration and diagnostic software for Ethernet networks in the automation environment (Factory Manager) | FL SWT | 2831044 | 1 |

## 5.2.4 Documentation

| Description | Type | Order No. | Pcs./Pkt. |
|---|---|---|---|
| "Configuring and installing the INTERBUS Inline product range" user manual | IB IL SYS PRO UM E | 2743048 | 1 |
| "Automation terminals of the Inline product range" user manual | IL SYS INST UM E | 2698737 | 1 |
| Quick start guide, PCWorx Version 5.00 or later | UM QS EN PC WORX EXPRESS | – | 1 |
| Quick start guide, PC Worx Version 6.00 or later | UM QS EN PC WORX | – | 1 |

| Description | Type | Order No. | Pcs./Pkt. |
|---|---|---|---|
| "PROFINET basics" user manual | UM EN PROFINET SYS | – | 1 |
| "PROFINET IO controller/device functions" user manual | UM EN PROFINET CTRL DEV | – | 1 |
| "Applicative system redundancy" application note | AH EN APPLICATIVE SYSTEM REDUNDANCY | – | 1 |

# A Appendix: Service and maintenance

## A 1 Error causes and remedies

Table A-1 Installation error causes and remedies

| Error | Cause | Remedy |
|---|---|---|
| No outputs can be set. | The voltage supply $U_S$ is absent (see diagnostic LED). | Connect the voltage supply. |
| The devices connected to the Inline controller cannot be read. | The voltage supply $U_S$ is absent (see diagnostic LED). | Connect the voltage supply. |
| The IEC 61131 program is not running. | Mode selector switch in STP position. | Set mode selector switch to RUN/PROG position. |
| The serial interface is not operating. | The connector pin assignment of the programming cable or of the connector adapter used is incorrect. | Use the PRG CAB MINI DIN programming cable, Order. No. 2730611, for the Inline controller. |
| The devices on the remote bus cannot be started up. | The IBS IL 24 RB-T (-PAC) Inline terminal is not installed directly after the Inline controller. | Insert the terminal as the first Inline terminal directly next to the Inline controller. |
| The diagnostic LED of a device is flashing quickly. | The device has not been assembled correctly. | Check the connection to the previous module (bus contacts). |
| ILC 170 ETH 2TX/ILC 190 ETH 2TX The FR and FF LEDs of the are flashing. | The ILC 170 ETH 2TX/ ILC 190 ETH 2TX cannot be addressed/the parameterization memory (SD card) is not inserted. | Insert a valid parameterization memory (SD card) that has been approved by Phoenix Contact into the slot of the ILC 170 ETH 2TX/ILC 190 ETH 2TX (see Figure 2-11 on page 2-16). |
| The Inline controller cannot be addressed via the IP address. | The IP address on the Inline controller has been changed or does not match the IP settings of the network devices involved. | Check the IP settings and adapt them to your application, if necessary (see also Section “Assigning the IP address for the controller/ BootP server” on page 3-2). |

## A 2 Updating the Inline controller firmware

The firmware (integrated software on the Inline controller) can be updated using the Ethernet interface. Such firmware updates are used exclusively for the addition of new functions that are implemented within the scope of continuous product improvement. No firmware update is required for normal system operation.

To update the firmware, please proceed according to the "Firmware update ÷ 4.6F/1.13" application note. It can be downloaded at www.phoenixcontact.net/catalog.

# A 3 Connecting unshielded cables

Unshielded cables are used to connect the I/O devices and the supply voltage to the Inline controller.

Connect these cables to the relevant Inline connectors using the spring-cage connection method. You may connect cables with diameters of 0.2 $mm^2$ to 1.5 $mm^2$ (24 - 16 AWG).

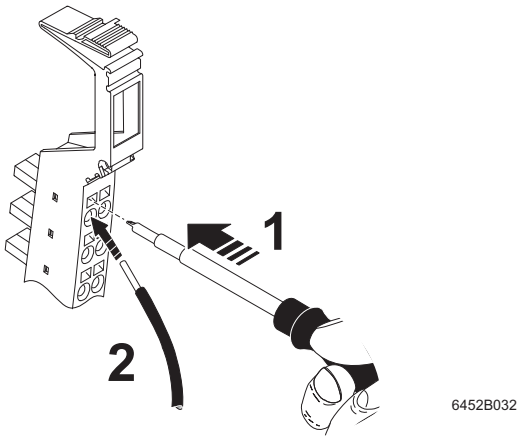



Figure A-1 Connecting unshielded cables

Wire the connectors according to your application.

When wiring, proceed as follows:

- Strip 8 mm off the cable.

Inline wiring is normally done without ferrules. However, it is possible to use ferrules. If using ferrules, make sure they are properly crimped.

- Push a screwdriver into the slot for the appropriate connection (Figure A-1, 1), so that you can plug the wire into the spring opening.
  Phoenix Contact recommends the SFZ screwdriver 1 - 0.6x3.5 (Order No. 12 04 51 7).

- Insert the wire (Figure A-1, 2). Remove the screwdriver from the opening. This clamps the wire.

After installation, the wires and the terminal points should be labeled.

# B Index


## A

Applicative system redundancy ..... 2-6

## B

Basic wiring of an output ..... 2-31
BootP ..... 3-2, 3-3
BootP server ..... 2-21

## C

Communication path ..... 2-21
Connecting the cables ..... A-2
Connecting the supply voltage ..... 2-27
Connecting unshielded cables ..... A-2
Connection elements ..... 2-9
Connector ..... 2-19
Control cabinet ..... 2-18
  See also Terminal box

## D

Diagnostic indicators ..... 2-11
Digital inputs ..... 2-30
DIN rail ..... 2-18

## E

Electronics base ..... 2-19
End clamp ..... 2-18
End plate ..... 2-18
Error causes ..... A-1
Error diagnostics ..... 2-11
Ethernet ..... 2-22
Ethernet interface ..... 2-22

## F

Fields of application
  ILC 130 ETH ..... 2-3
  ILC 150 ETH, ILC 155 ETH, ILC 170 ETH 2TX, ILC 190 ETH 2TX ..... 2-3
  ILC 170 ETH 2TX, ILC 190 ETH 2TX ..... 2-6
FTP functions ..... 3-12

## H

Hardware requirements ..... 1-1

## I

ILC 130 ETH, ILC 150 ETH, ILC 155 ETH
  Connectors ..... 2-9
  Diagnostic and status indicators ..... 2-9
  Electronics base ..... 2-9
  End plate ..... 2-9
  Ethernet connection ..... 2-9
  Mode selector switch ..... 2-9
  Parameterization memory, integrated ..... 2-15
  Reset button ..... 2-9
  V.24 (RS-232) interface ..... 2-9
ILC 170 ETH 2TX, ILC 190 ETH 2TX
  Card holder ..... 2-10
  Connector ..... 2-10
  Diagnostic and status indicators ..... 2-10
  Electronics base ..... 2-10
  End plate ..... 2-10
  Ethernet connection ..... 2-10
  Mode selector switch ..... 2-10
  Parameterization memory, pluggable ..... 2-15
  Reset button ..... 2-10
  RS-232 interface ..... 2-10
  Slot for the parameterization memory ..... 2-10
Inline local bus ..... 2-3
INTERBUS ..... 2-25
IP address
  Setting the ..... 3-2, 3-4
IP20 ..... 2-18

## L

Local bus ..... 2-25

## M

MAC address ..... 2-21
Mode selector switch ..... 2-14
Mounting ..... 2-18, 2-19
Mounting location ..... 2-18
Mounting position ..... 2-18

## O

Operating elements ..... 2-9

## P

Parameterization memory ..... 2-15
Parameterization memory, integrated ..... 2-15
Parameterization memory, pluggable..... 2-15
Power supply ..... 2-26
Power supply without fall-back characteristic curve 2-26, 5-1
PROFINET
"Read PROFINET" dialog box ..... 3-9
Integrating the IO device (example)..... 3-8
IO device inserted ..... 3-10
IO device process data..... 3-10
PROFINET IO device ..... 3-5
Reading in PROFINET ..... 3-8
Switching on the IO device function..... 3-5

## R

Removal..... 2-18, 2-19
Replacement..... 2-21
Reset button..... 2-15
Resetting the Inline controller ..... 2-15
Resetting the controller ..... 3-6

## S

Serial PRG interface..... 2-23
Sizing of the power supply ..... 2-26
Software requirements..... 1-1
System variables..... 4-1

## T

Terminal box ..... 2-18

## U

Updating the firmware ..... A-1

## V

V.24 (RS-232) cable..... 2-23
Voltage supply
Connecting the ..... 2-27